LARGE-SCALE SCHEDULING OPTIMIZATION CONSIDERING UNCERTAINTY: REFINERY OPERATIONS

by

ZHENYA JIA

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

Large-Scale Scheduling Optimization Considering Uncertainty: Refinery Operations

by ZHENYA JIA

Dissertation Advisor: Dr. Marianthi Ierapetritou

This thesis addresses the optimization of large-scale scheduling problem, specifically in the area of refinery operations. Due to the high complexity of the the entire system, the refinery scheduling problem is spatially decomposed into three subproblems: the crude-oil unloading and blending, the production-unit operations and the product blending and delivery. Comprehensive mathematical models are developed to describe the characteristics of each of those subproblems and solved efficiently due to the full untilization of time continuity.

An important consideration in scheduling and of particular interest to refinery operations is the issue of uncertainty introduced due to variability in product demands, prices, raw material availabilities, and process parameters. A rigorous methodology is thus developed that can be utilized to address the problem of uncertainty in decision-making process. The proposed approach is based on sensitivity analysis within a branch-and-bound solution methodology that results in the identification of the ranges of uncertain parameters where the schedules remain feasible and optimal, and the determination of a set of alternative schedules to cover the uncertainty space. In order to obtain schedules that are less sensitive to the model data, we developed a multiobjective robust optimization model to capture the trade-offs of economic optimality and decision robustness in the face of uncertainty and enable the generation of a number of alternative solutions.

These uncertainty analysis approaches are applied to the optimal operations of crude oil unloading and mixing, and gasoline blending and distribution to illustrate the importance of considering uncertainty in refinery scheduling operations.

ACKNOWLEDGMENT

I whole-heartedly acknowledge the efforts of Prof. Marianthi G. Ierapetritou, my adviser, whose encouragement and constructive criticism, striking a perfect balance between providing direction and encouraging independence throughout the course of the research, served as a constant source of inspiration. I also extend my thanks to my committee members Prof. David Coit, Prof. Ioannis P. Androulakis and Prof. Henrik Pedersen for their guidance and technical advice.

I would like to express my gratitude to other members of this group and other graduate students of the department with whom I shared enjoyable moments. My special thanks to Vishal and Ipsita in whose company I have enjoyed the warmth of family, hours of fruitless banter of friends and healthy competition of peers.

The financial support of National Science Foundation under the NSF CAREER program CTS-9983406 during the period of research is gratefully appreciated.

Finally, I present my most sincere thanks to my family for their constant support and encouragement.

DEDICATION

To my parents and my brother

Ruichun Jia and Qingmei Li

Zhenyu Jia

Contents

List of Figures

Chapter 1

Introduction

Oil refineries are increasingly concerned with improving the planning of their operations and optimizing not only single production units but the whole production enterprise.

To address the problem of scheduling of refinery operations, since the commercial tools are lacking the flexibility to handle plant particularities, refineries are developing in-house tools strongly based on simulation[108] to enhance the knowledge of the system. Specific applications of refinery scheduling models that appear in the literature include the work of Bodington[16], Rigby et al.[91], Lee et al.[65], Shah[103]. More recently Joly et al.[60], presented a series of mathematical programming models to address a variety of problems in refinery operations. Song et al.[107], incorporate environmental impacts as part of the tradeoffs that have to be considered in decision-making process in refinery operations. The case of Kuwait refinery is considered in Lababidi et al.[64], where a simplified flowsheet is optimized within an advanced control-planning environment. Li et al.[66], address the problem of crude oil unloading, storage and processing using a MINLP model. In the product distribution side, scheduling of pipeline is considered in the work of Rejowski and Pinto[89], whereas the product movement and scheduling of oil refinery is considered by Paolucci et al.[79]. A review of process technology in the petroleum refining industry[74] identifies the need of efficient scheduling models as the most urgent need in oil companies given the expected benefits. Furthermore, the modeling of the overall refinery operation from the crude oil arrival to the distribution of oil products gives rise to intractable mathematical models. Thus, decomposition methodologies have long been recognized as the major avenue to overcome this computational burden.

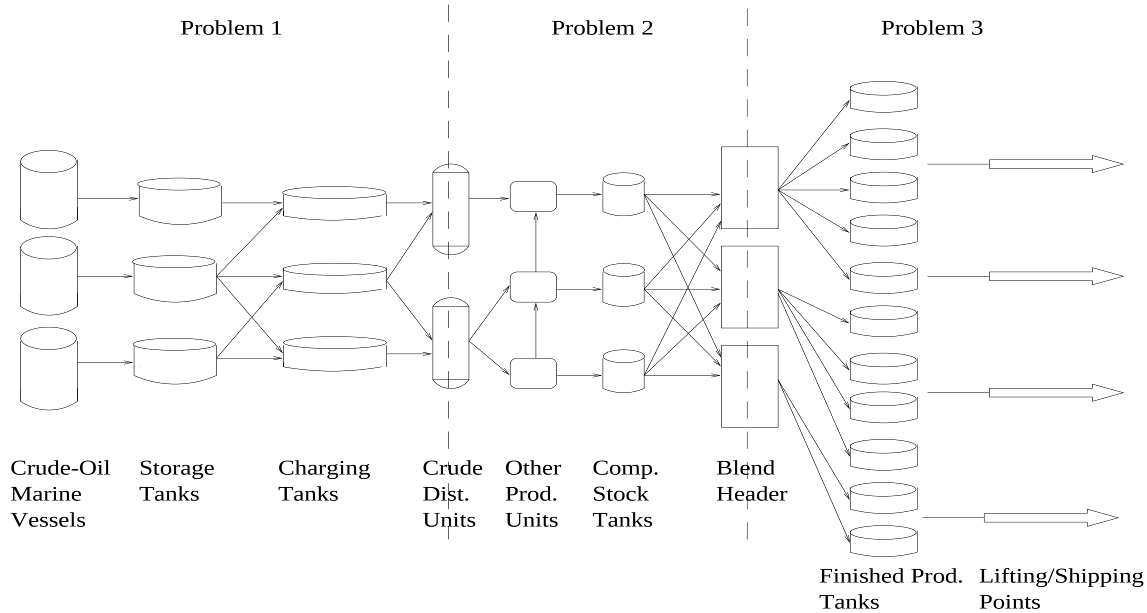


Figure 1.1: Graphic Overview of Refinery System

Substantial benefits can be achieved through the use of optimization techniques in plant operations by improving the resource utilization at different levels of decision-making process as illustrated for refinery operations by the work reviewed above. However, uncertainty exists in realistic manufacturing environment due to lack of accurate process models and variability of process and environment data. The presence of unertainty can substantially reduce or eliminate the advantages of optimization approaches. Therefore, it is of great importance to develop systematic methods to address the problem of scheduling under uncertainty. Existing work mainly includes stochastic programming approaches involving chance constraints and two-stage programming[18,59], as well as robust optimization methods[13,68,111]. The issue of uncertainty has been extensively addressed in operations research and chemical engineering literature. A recent thorough review of the subject is provided by Sahinidis[96]. The main findings of this review are the lack of efficient rigorous methodology that can be utilized to address the problem of uncertainty providing the decision maker with competing alternative solutions.

In this work, the optimization of an entire refinery operations as illustrated in Figure 1.1 is addressed. The thesis os organized as follows. In the first four chapters the problem of scheduling of refinery operations is addressed first whereas chapters 5, 6 and 7 are devoted to the issue of uncertianty consideration. Chapter 2 studies the first subproblem of crude-oil short-term scheduling, which involves the optimal operation of crude-oil unloading from vessels, its transfer to storage tanks and the charging schedule for each crude oil mixture to the distillation units. The mathematical model follows a continuous time representation and thus results in a reduction of the number of variables and constraints compared to discrete time modeling. It is shown that the proposed formulation outperformed the published results at least by an order of magnitude in terms of the CPU time, but more importantly in terms of the number of binary, continuous variables and constraints. Thus it results in a much easier problem to solve as suggested by the smaller number of nodes required for the solution of the MILP model using the Branch and Bound solution procedure. The mathematical model of the second part of refinery system is presented in chapter 3, which includes three processing stages: extraction, dewaxing and hydrofinishing. Chapter 4 addresses the optimal operation of gasoline blending and distribution. The mathematical model developed involves material balances of each product at each tank, the timing requirements that enforce the correct sequence and duration of order implementation, storage constraints to enforce the tank capacities, the requirement to satisfy all the orders at the due dates, and logical constraints that secure modeling integrity in terms of binary variables.

In chapter 5, a robust optimization approach is developed to obtain schedules that are less sensitive to the model data and determine a set of Pareto optimal solutions that can capture the trade-off among the various conflicting objectives. In order to incorporate both solution and model robustness in scheduling problems, we utilized the concept of upper partial mean[2] to measure the positive deviation and introduced slack variables in the demand constraints to evaluate the model infeasibility. In addition to the original objective, the solution robustness and model robustness are also considered in the objective function. Therefore, the original deterministic model is reformulated into a multiobjective optimization problem. Normal Boundary Intersection (NBI) is a technique which has a variety of advantages to obtain the Pareto surface, and it is utilized to efficiently generate evenly spread Pareto optimal points, each of which represents a trade-off solution between the expected objective value over all scenarios, feasibility of the schedule and deviation to the expectation. In chapter 6, a systematic framework is presented to address the problem of accounting for uncertainty in the scheduling decision-making process. The deterministic scheduling problem is solved at the nominal values using a branch-and-bound solution approach, and the dual information is collected at each leaf node. The inference-based sensitivity analysis[27] is utilized to determine the range of objective function for a given perturbation on the operating parameters. The relaxed LPs of the leaf nodes are solved and the nodes with objective values with the predicted range are saved. The branch-and-bound procedure is continued on those remaining leaf nodes so as to determine a set of alternative schedules that cover the uncertainty range. The alternative schedules are evaluated based on the average and nominal performance of the objective function (makespan or profit), as well as the standard deviation. This approach uncovers the detailed information about alternative schedules under parameter variability without drastically increasing the computational complexity of the solution scheme compared with the deterministic model. Chapter 7 presents a systematic framework to solve the parametric mixed integer linear programming (pMILP) problems where uncertain parameters are present on the right hand side (RHS) of the constraints. It mainly consists of two steps: LP/mpLP sensitivity analysis and updating the B&B tree. For the case of multiple uncertain parameters, a novel algorithm of multiparametric linear programming (mpLP) is proposed that does not require the construction of the LP tableaus but solves a set of NLP problems iteratively using the commercially available global optimization solver BARON. At each iteration, a point at which the objective value cannot be represented by the current optimal functions is determined and the new optimal function is included in the next iteration. Given the range of uncertain parameters, the output of this proposed framework is a set of optimal integer solutions and their corresponding critical regions and optimal functions. A number of examples are presented to illustrate the applicability of the proposed approach and comparison with existing methods.

The last chapter addresses the implementation of the proposed uncertainty analysis approaches to the refinery scheduling problems considering demand variability. In particular, the crude oil unloading and mixing, and gasoline blending and distribution problems as presented in chapter 2 and 4 are studied. The multiobjective robust optimization appproach proposed in chapter 5 is employed to solve the multiobjective model to generate the Pareto optimal surface that captures the trade-off between the economic optimality and solution robustness. The parametric MILP approach described in chapter 7 is utilized to solve the corresponding pMILP model of the refinery scheduling problem, so as to determine the optimal operation schedule and objective value with respect to uncertain parameters.

Chapter 2

Crude Oil Operations

2.1 Introduction

In the literature, mathematical programming technologies have been extensively concerned and developed in the area of long-term refinery planning[17,88], while short-term scheduling has received less attention.

Refinery planning optimization is mainly addressed through successive linear programming approach, such as GRTMPS (Haverly Systems), PIMS (Aspen Technology) and RPMS (Honeywell Hi-Spec Solutions), while more rigorous nonlinear planning models for refinery production were developed recently[73,84] . Scheduling has been mainly addressed for batch plants. Extensive reviews can be found in Reklaitis[90], Pinto and Grossmann[83], Ierapetritou and Floudas[51]. Continuous plants, however have received less attention in the open literature concerning the scheduling optimization problem. Sahinidis and Grossmann[97] considered the problem of cyclic scheduling of multiproduct continuous plants for the single stage case, and by Pinto and Grossmann[82] for the multistage case (typical of lube-oil production). Ierapetritou and Floudas[52] extended their batch scheduling model to consider continuous and mixed production facilities. Mathematical programming techniques were applied to scheduling of crude-oil handling at the refinery by Shah[103] and Lee et al.[65] that presented MILP models for the detailed short-term scheduling of crude oil unloading, based on time discretization. The resultant models are prohibitively expensive to solve due to the discretization of time.

A somewhat related study on the allocation of crude-oils to storage tanks to enhance the flexibility of the crude-oil mixing operation is presented in Kelly and Forbes[61]. This describes the rules necessary to decide which crude-oils should go to which storage tanks when there are more crude-oils than tanks (i.e., the classic pigeon hole principle problem). Zhang and Zhu[114] proposed a decomposition approach, which decomposes the overall refinery model into a site level and a process level. The site level determines operating rules for each process and the process level returns updated performance to the site level for further optimization. This procedure continues until a specified tolerance is met.

The refinery scheduling problem considered in this chapter includes the unloading from the vessels to the storage tanks, transfers from storage tanks to the charging tanks and charging to the crude-oil distillation units. The objective of the scheduling procedure is to minimize the total cost of operation. The state-task network (STN) representation introduced by Kondili et al.[63] is used throughout this chapter.

This chapter considers the first part of the overall picture for optimization of oil-refinery opterations as depicted in Figure 1.1. It involves the unloading from the vessels to the storage tanks, transfers from storage tanks to the charging tanks and charging to the crude-oil distillation units. The objective of the scheduling procedure is to minimize the total cost of operation. This chapter is organized as follows. Section 2.2 states the problem of crude oil unloading with inventory control. In section 2.3, the mathematical formulation is presented and then applied to four case studies in the following section. Efficient solution is achieved mainly due to the exploitation of the continuous-time nature of the problem.

2.2 Problem Definition

A typical crude-oil unloading system considered here consists of crude-oil marine vessels - used for crude-oil transportation, storage tanks - serving as the off-loading tanks for incoming crude-oils, charging tanks - used for crude-oil blending, and crude-oil distillation units - where the main hydrocarbon separation into downstream feedstocks takes place, as illustrated in Figure 1.1. Crude-oil vessels unload crude oil into storage tanks after arrival at the refinery docking station. Then the crude-oil is transferred from storage tanks to charging tanks, in which a crude-oil mix is produced. The crude-oil mix in each charging tank may then be charged into one or more crude-oil distillation units. The following information will be given: a) arrival times of marine vessels; b) capacity limitations of tanks; c) key component concentration ranges; d) time horizon under consideration.

The problem is to determine the following variables: a) waiting time of each vessel at sea; b) unloading duration times from vessels to storage tanks; c) crude oil transfer duration times from storage tanks to charging tanks; d) crude-oil mix charging duration times from charging tanks to CDU; e) opening inventory and concentration levels in storage and charging tanks, so as to minimize the operating costs.

The operating rules that have to be followed are: a) a vessel has to wait at sea if a preceding vessel doesn't leave the docking station; b) if crude-oil is fed into the charging tank, the charging tank cannot charge the CDU simultaneously and vice versa (sometimes referred to as standing-gauge operation); c) each charging tank can only charge one CDU at one time interval; d) each CDU can only be charged by one charging tank at one time interval; e) CDU must be operated continuously throughout the time horizon of scheduling.

The objective is to minimize the total operation cost, which includes sea waiting cost, unloading cost and inventory cost. Based on the above description of the problem, the following mathematical formulation is proposed for the efficient solution of the problem.

2.3 Mathematical Formulation

The following assumptions are proposed: a) the times required for CDU mode change are neglected; b) perfect mixing is assumed in the tanks; c) the property state of each crude-oil or mixture is decided only by specific key components and finally no explicit splitting operations are performed.

The mathematical model involves mainly material balance constraints, allocation constraints, sequence constraints, crude-oil supply and crude-oil mix demand constraints. Material balance constraints connect the amounts of material in one unit at one event point to that at the next event point. Allocation constraints set the delivery assignments between the vessels, storage tanks, charging tanks and the CDUs, and the beginning and finishing times of each operation are determined by the sequence constraints. The supply constraints ensure that all of the vessel's cargo is unloaded and the demand constraints ensure that all the demands of crude-oil mix will be satisfied during the time horizon.

Material Balance Constraints for Vessel (v)

Each vessel (v) initially has (Vv0(v)) crude oil and unloads all its crude-oil to storage tanks before leaving the docking station:

$$Vv0(v) = \sum_{i \in I_v} \sum_{n} Bv(v,i,n), \forall v \in V \tag{2.1a}$$

The volume of crude-oil in vessel (v) at event point (n) (Vv(n)) should be equal to that at event point (n-1) minus the total volume of crude-oil being unloaded from vessel (v) to storage tanks at the current event point (n) ($\sum_n Bv(v,i,n)$).

$$Vv(v,n) = Vv0(v) - \sum_{i \in I_v} Bv(v,i,n), \forall v \in V, n = 1 \tag{2.1b}$$

$$Vv(v,n) = Vv(v,n-1) - \sum_{i \in I_v} Bv(v,i,n), \forall v \in V, n \in N, n \neq 1 \tag{2.1c}$$

Material Balance Constraints for Storage Tank (i)

Each storage tank (i) initially has (Vs0(i)) crude-oil. The inventory level of crude oil in tank (i) at event point (n) is equal to that at event point (n-1) adjusted by any amounts fed from vessels or transferred to charging tanks.

$$Vs(i,n) = Vs0(i) + \sum_{v \in V_i} Bv(v,i,n) - \sum_{j \in J_i} Bs(i,j,n), \forall i \in I, n = 1 \quad (2.2a)$$

$$Vs(i,n) = Vs(i,n-1) + \sum_{v \in V_i} Bv(v,i,n) - \sum_{j \in J_i} Bs(i,j,n), \forall i \in I, n \in N, n \neq 1 \quad (2.2b)$$

Material Balance Constraints for the Charging Tank (j)

Each charging tank (j) contains (Vb0(i)) crude-oil at the start. The inventory level of crude-oil mix in charging tank (j) at event point (n) is equal to that at event point (n-1) adjusted by any amounts transferred from storage tanks or charged to CDUs.

$$Vb(j,n) = Vb0(j) + \sum_{i \in I_j} Bs(i,j,n) - \sum_{l \in L_j} Bb(j,l,n), \forall j \in J, n = 1 \quad (2.3a)$$

$$Vb(j,n) = Vb(j,n-1) + \sum_{i \in I_j} Bs(i,j,n) - \sum_{l \in L_j} Bb(j,l,n), \forall j \in J, n \in N, n \neq 1 \quad (2.3b)$$

Material Balance Constraints for the Component (k) in Storage Tank (i)

Based on the assumption that the property of each crude-oil is decided by key components, such as sulfur or metals, in the crude-oil, the appropriate material balance constraints for the components are formulated to track the quantities and restrain the concentrations from exceeding pre-defined specifications. The initial or opening amount of component (k) in storage tank (i) can be determined by [the initial amount of crude-oil in tank (i)] * [the initial concentration of component (k) in tank (i)]. The amount of component (k) in storage tank (i) at event point (n) is equal to that at event point (n-1) adjusted by any amounts unloaded from vessels or transferred to the charging tanks.

$$Vss(i,k,n) = Vs0(i) * Ds0(i,k) + \sum_{v \in V_i} Bv(v,i,n) * Dv(v,k) - \sum_{j \in J_i} Bss(i,j,k,n), \quad \forall i \in I, k \in K, n = 1 \tag{2.4a}$$

$$Vss(i,k,n) = Vss(i,k,n-1) + \sum_{v \in V_i} Bv(v,i,n) * Dv(v,k) - \sum_{j \in J_i} Bss(i,j,k,n), \quad \forall i \in I, k \in K, n \in N, n \neq 1 \tag{2.4b}$$

Constraint (2.4c) and (2.4d) express the requirement that the concentration of component (k) in storage tank (i) should be in the required concentration range.

$$Vs(i,n) * Dsmin(i,k) \leq Vss(i,k,n) \leq Vs(i,n) * Dsmax(i,k), \quad \forall i \in I, k \in K, n \in N \tag{2.4c}$$

$$Bs(i,j,n) * Dsmin(i,k) \leq Bss(i,j,k,n) \leq Bs(i,j,n) * Dsmax(i,k), \quad \forall i \in I, j \in J_i, k \in K, n \in N \tag{2.4d}$$

Material Balance Constraints for the Component (k) in Charging Tank (j)

Similar to constraints (2.4a) - (2.4d), following contraints represent the material balance for the components in charging tanks. Note that in constraint (2.5b), $\sum_{i \in I_j} Bss(i,j,k,n)$ can be written as $\sum_{i \in I_j} Bs(i,j,n) * Ds0(i,k)$ in the case that the concentration of components in the storage tanks is constant.

$$Vbb(j,k,n) = Vb0(j) * Db0(j,k) + \sum_{i \in I_j} Bss(i,j,k,n) - \sum_{l \in L_j} Bbb(j,l,k,n), \quad \forall j \in J, k \in K, n = 1 \tag{2.5a}$$

$$Vbb(j,k,n) = Vbb(j,k,n-1) + \sum_{i \in I_j} Bss(i,j,k,n) - \sum_{l \in L_j} Bbb(j,l,k,n), \quad \forall j \in J, k \in K, n \in N, n \neq 1 \tag{2.5b}$$

$$Vb(j,n) * Dbmin(j,k) \leq Vbb(j,k,n) \leq Vb(j,n) * Dbmax(j,k), \quad \forall j \in J, k \in K, n \in N \tag{2.5c}$$

$$Bb(j,l,n) * Dbmin(j,k) \leq Bbb(j,l,k,n) \leq Bb(j,l,n) * Dbmax(j,k), \quad \forall j \in J, l \in L_j, k \in K, n \in N \tag{2.5d}$$

Components Concentration Consideration

The following bilinear equations are required to ensure consistent balance of the outgoing component streams at event point (n).

$$Bss(i,j,k,n) = Bs(i,j,n) * Ds0(i,k), \forall i \in I, j \in J_i, k \in K, n = 1 \tag{2.6a}$$

$$Bss(i,j,k,n) * Vs(i,n-1) = Bs(i,j,n) * Vss(i,k,n-1), \quad \forall i \in I, j \in J_i, k \in K, n \in N, n \neq 1 \tag{2.6b}$$

$$Bbb(j,l,k,n) = Bb(j,l,n) * Db0(j,k), \forall j \in J, l \in L_j, k \in K, n = 1 \tag{2.6c}$$

$$Bbb(j,l,k,n) * Vb(j,n-1) = Bb(j,l,n) * Vbb(j,k,n-1), \quad \forall j \in J, l \in L_j, k \in K, n \in N, n \neq 1 \tag{2.6d}$$

Capacity Constraints

Constranits (2.7a) and (2.7b) express the volume capacity limitations for storage and charging tanks.

$$Vs(i,n) \leq Vsmax(i), \forall i \in I, n \in N \tag{2.7a}$$

$$Vb(j,n) \leq Vbmax(j), \forall j \in J, n \in N \tag{2.7b}$$

Flow Rate Constraints

The following constraints express the limitations of minimum and maximum flow rate when a crude-oil or a mix is being transferred. The volume of crude-oil or mix being transferred should be between the limits of [duration time * fmin] and [duration time * fmax].

$$(Tvf(v,i,n) - Tvs(v,i,n)) * fmin \leq Bv(v,i,n) \leq (Tvf(v,i,n) - Tvs(v,i,n)) * fmax, \quad \forall v \in V, i \in I_v, n \in N \tag{2.8a}$$

$$(Tsf(i,j,n) - Tss(i,j,n)) * fmin \leq Bs(i,j,n) \leq (Tsf(i,j,n) - Tss(i,j,n)) * fmax, \quad \forall i \in I, j \in J_i, n \in N \tag{2.8b}$$

$$(Tbf(j,l,n) - Tbs(j,l,n)) * fmin \leq Bb(j,l,n) \leq (Tbf(j,l,n) - Tbs(j,l,n)) * fmax, \quad \forall j \in J, l \in L_j, n \in N \tag{2.8c}$$

Allocation Constraints

At most one CDU (l) can be charged by charging tank (j) at one time and vice versa and the CDU must be in continuous operation.

$$\sum_{j \in J_l} z(j,l,n) \leq 1, \forall l \in L, n \in N \tag{2.9a}$$

Charging tank (j) cannot charge CDU and be fed by storage tank at the same time.

$$y(i,j,n) + \sum_{l \in L_j} z(j,l,n) \leq 1, \forall j \in J, n \in N \tag{2.9b}$$

The following constraints force binary variables x(v,i,n), y(i,j,n) and z(j,l,n) to be 1 if Bv(v,i,n), Bs(i,j,n), and Bb(j,l,n) are not zero, respectively, otherwise, they are equal to zero. These constraints are required in order to maintain the one-to-one assignment of the amounts being transferred and the corresponding binary variables.

Thus the current allocation constraints can be enforced.

$$x(v,i,n) * Vmin \leq Bv(v,i,n) \leq x(v,i,n) * Vmax, \forall v \in V, i \in I_v, n \in N \quad (2.9c)$$

$$y(i,j,n) * Vmin \leq Bs(i,j,n) \leq y(i,j,n) * Vmax, \forall i \in I, j \in J_i, n \in N \quad (2.9d)$$

$$z(j,l,n) * Vmin \leq Bb(j,l,n) \leq z(j,l,n) * Vmax, \forall j \in J, l \in L_j, n \in N \quad (2.9e)$$

Demand Constraints

The demand of crude-oil mix (j) should be met by the total amount of crude-oil mix being lifted from a charging tank (j).

$$\sum_{l \in L_j} \sum_{n \in N} Bb(j,l,n) = DM(j), \forall j \in J \quad (2.10)$$

Sequence Constraints: Vessel (v) $\longrightarrow$ Storage Tank (i)

The requirement that each vessel can start unloading crude-oil only after its arrival and must empty its cargo before the end of the time horizon is expressed through constraint (2.11a) and (2.11b) respectively.

$$Tvs(v,i,n) \geq Tarr(v) * x(v,i,n), \forall v \in V, i \in I_v, n \in N \quad (2.11a)$$

$$Tvf(v,i,n) \leq H, \forall v \in V, i \in I_v, n \in N \quad (2.11b)$$

Each vessel (v) starting to unload to tank (i) at event point (n+1) should be after the end of unloading at event point (n).

$$Tvs(v,i,n+1) \geq Tvf(v,i,n) - H * (1 - x(v,i,n)),$$
$$\forall v \in V, i \in I_v, n \in N, n \neq NE \quad (2.11c)$$

$$Tvs(v,i,n+1) \geq Tvs(v,i,n), \forall v \in V, i \in I_v, n \in N, n \neq NE \quad (2.11d)$$

$$Tvf(v,i,n+1) \geq Tvf(v,i,n), \forall v \in V, i \in I_v, n \in N, n \neq NE \quad (2.11e)$$

If vessel (v') arrives at the docking station later than vessel (v), then it cannot start unloading until vessel (v) finishes and leaves.

$$\sum_n Tvst(v', i, n) \geq \sum_n Tvft(v, i, n), \forall v \in V_i, v' \in V_i, i \in I, Tarr(v') > Tarr(v) \tag{2.11f}$$

Similar sequence constraints are imposed for the crude-oil transfer between storage tank (i) and charging tank (j) [constraints (2.12a) - (2.12d)] and charging tank (j) and CDU (l) [constraints (2.13a) - (2.13d)].

Sequence Constraints: Storage Tank (i) $\longrightarrow$ Charging Tank (j)

$$Tss(i, j, n+1) \geq Tsf(i, j, n) - H * (1 - y(i, j, n)),$$
$$\forall i \in I, j \in J_i, n \in N, n \neq NE \tag{2.12a}$$

$$Tss(i, j, n+1) \geq Tss(i, j, n), \forall i \in I, j \in J_i, n \in N, n \neq NE \tag{2.12b}$$

$$Tsf(i, j, n+1) \geq Tsf(i, j, n), \forall i \in I, j \in J_i, n \in N, n \neq NE \tag{2.12c}$$

$$Tsf(i, j, n) \leq H, \forall i \in I, j \in J_i, n \in N \tag{2.12d}$$

Sequence Constraints: Charging Tank (j) $\longrightarrow$ CDU (l)

$$Tbs(j, l, n+1) \geq Tbf(j, l, n) - H * (1 - z(j, l, n)),$$
$$\forall j \in J, l \in L_j, n \in N, n \neq NE \tag{2.13a}$$

$$Tbs(j, l, n+1) \geq Tbs(j, l, n), \forall j \in J, l \in L_j, n \in N, n \neq NE \tag{2.13b}$$

$$Tbf(j, l, n+1) \geq Tbf(j, l, n), \forall j \in J, l \in L_j, n \in N, n \neq NE \tag{2.13c}$$

$$Tbf(j, l, n) \leq H, \forall j \in J, l \in L_j, n \in N \tag{2.13d}$$

Each charging tank (j) can either be fed from storage tank (i) or charge CDU (l) at any event point (n).

$$Tss(i,j,n+1) \geq Tbf(j,l,n) - H*(1-z(j,l,n)),$$
$$\forall i \in I, j \in J_i, l \in L_j, n \in N, n \neq NE \tag{2.13e}$$
$$Tbs(j,l,n+1) \geq Tsf(i,j,n) - H*(1-y(i,j,n)),$$
$$\forall i \in I, j \in J_i, l \in L_j, n \in N, n \neq NE \tag{2.13f}$$

Charging tank (j) should start charging CDU (l) after the completion of charging other CDUs in previous event points.

$$Tbs(j,l,n+1) \geq Tbf(j,l',n) - H*(1-z(j,l',n)),$$
$$\forall j \in J, l \in L_j, l' \in L_j, n \in N, n \neq NE \tag{2.13g}$$

Since each CDU (l) must be operated continuously, the total operation time of each CDU (l) should be equal to the time horizon (H). Constraints (2.13i) and (2.13j) express that if CDU (l) is charged at event point (n), then the next charge will start at the ending time of this event point (n).

$$\sum_n \sum_{j \in J_l} (Tbf(j,l,n) - Tbs(j,l,n)) = H, \forall l \in L, n \in N \tag{2.13h}$$
$$Tbs(j,l,n+1) \geq Tbf(j',l,n) - H*(1-z(j',l,n)),$$
$$\forall j \in J_l, j' \in J_l, l \in L, n \in N, n \neq NE \tag{2.13i}$$
$$Tbs(j,l,n+1) \leq Tbf(j',l,n) + H*(1-z(j',l,n)),$$
$$\forall j \in J_l, j' \in J_l, l \in L, n \in N, n \neq NE \tag{2.13j}$$

Beginning - Ending Time Consideration

The starting and end times of unloading of vessel (v) are essentially $Tvst(v,i,n) = Tvs(v,i,n) * x(v,i,n)$ and $Tvft(v,i,n) = Tvf(v,i,n) * x(v,i,n)$ that involve bilinear terms (continuous * binary). By applying Glover's transformation to constraints (2.14a) - (2.14d), linearity can be preserved. Constraints (2.14b) and (2.14d) are needed to force Tvst(v,i,n) and Tvft(v,i,n) to be zero when x(v,i,n) is zero.

$$Tvs(v,i,n) - H * (1 - x(v,i,n)) \leq Tvst(v,i,n) \leq Tvs(v,i,n),$$
$$\forall v \in V, i \in I_v, n \in N \quad (2.14a)$$

$$Tvst(v,i,n) \leq H * x(v,i,n), \forall v \in V, i \in I_v, n \in N \quad (2.14b)$$

$$Tvf(v,i,n) - H * (1 - x(v,i,n)) \leq Tvft(v,i,n) \leq Tvf(v,i,n),$$
$$\forall v \in V, i \in I_v, n \in N \quad (2.14c)$$

$$Tvft(v,i,n) \leq H * x(v,i,n), \forall v \in V, i \in I_v, n \in N \quad (2.14d)$$

Objective Function

The objective of this problem is to minimize the total operating cost.

$$\begin{aligned} cost = {} & Csea * \sum_v \sum_{i \in I_v} \sum_n (Tvst(v,i,n) - Tarr(v)) \\ & + Cunload * \sum_v \sum_{i \in I_v} \sum_n (Tvft(v,i,n) - Tvst(v,i,n)) \\ & + Cinvst * H * \sum_i (\sum_n Vs(i,n) + Vs0(i))/(NE+1) \\ & + Cinvbi * H * \sum_j (\sum_n Vb(j,n) + Vb0(j))/(NE+1) \\ & + Cset * (\sum_j \sum_{l \in L_j} \sum_n z(j,l,n) - NST) \end{aligned} \quad (2.15)$$

The first term in the objective function is the sea waiting cost, where $\sum_{v,i \in I_v,n}(Tvst(v,i,n) - Tarr(v))$ is the total waiting time at sea for all the vessels. The second term represents the unloading cost, where $\sum_v \sum_{i \in I_v} \sum_n (Tvft(v,i,n) - Tvst(v,i,n))$ is the total unloading duration of all the vessels. The total inventory levels of storage tanks and charging tanks are approximated by $\sum_i(\sum_n Vs(i,n) + Vs0(i))/(NE+1)$ and $\sum_j(\sum_n Vb(j,n) + Vb0(j)/(NE+1)$ respectively, which corresponds to the average value of the inventory level of the tanks over the time horizon under consideration. This approximation is selected in order to maintain the linearity of the model. As it will be shown in the examples in the next section, this simplification corresponds to a close approximation of the actual inventory cost and results in reasonable schedules.

2.4 Case Studies: Results and Comparisons

Four examples are studied with the data obtained from Lee et al.(1996). The data for examples 1 - 4 are given in Table 2.1 - 2.4 respectively, while the crude-oil flow networks are illustrated by Figures 2.1 - 2.4. Example 1 deals with a small size problem, while Example 4 presents a problem with 3 vessels, 6 storage tanks and 4 charging tanks that constitute an industry size problem. Two key components concentrations are considered in Example 2. Example 3 has the same size as example 2, however, oil mixing occurs in storage tanks as well as in charging tanks. Hence the material balance constraints for the components in charging tanks will also be applied to storage tanks.

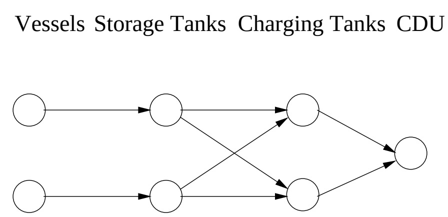


Figure 2.1: Oil flow network for Example 1.

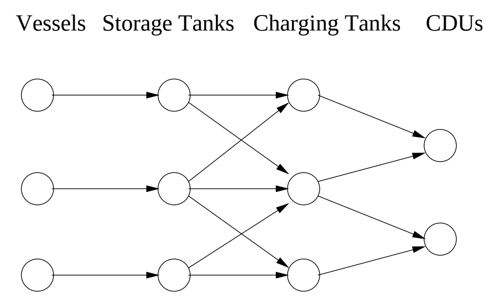


Figure 2.2: Oil flow network for Example 2.

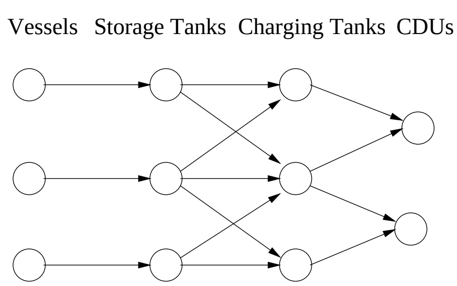


Figure 2.3: Oil flow network for Example 3.

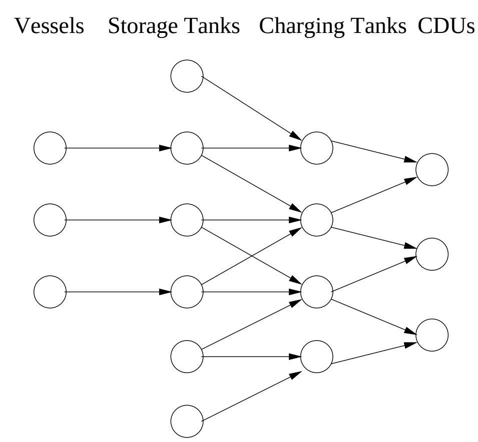


Figure 2.4: Oil flow network for Example 4.

As shown in Table 2.5, the proposed MILP formulation results in much smaller model in terms of constraints, continuous and binary variables. Note that for the

Scheduling Horizon(# of unit time)			8
Number of Vessel Arrivals			2
	Arrival Time	Amount of Crude	Concentration of key component
Vessel 1	1	100	0.01
Vessel 2	5	100	0.06
Number of Storage Tanks			2
Storage Tanks	Capacity	Initial Oil Amount	Concentration of key component
Tank 1	100	25	0.01
Tank 2	100	75	0.06
Number of Charging Tanks			2
Charging Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min, max)
Tank 1	100	50	0.02 (0.015, 0.025)
Tank 2	100	50	0.05 (0.045, 0.055)
Number of CDU			1
Unit costs involved in vessel operation			Unloading cost: 8, Sea waiting cost: 5
Tank inventory unit costs			Storage tank: 0.08, Charging tank: 0.05
Unit changeover cost for charged oil switch			50 (independent of sequence and CDU)
Demand of mixed oils by CDUs			Oil mix 1: 100, Oil mix 2: 100

Table 2.1: System Information for Example 1

Scheduling Horizon(# of unit time)			10	
Number of Vessel Arrivals			3	
	Arrival Time	Amount of Crude	Component 1	Component 2
Vessel 1	1	100	0.01	0.04
Vessel 2	4	100	0.03	0.02
Vessel 3	7	100	0.05	0.01
Number of Storage Tanks			3	
Storage Tanks	Capacity	Initial Oil Amount	Component 1	Component 2
Tank 1	100	20	0.01	0.04
Tank 2	100	50	0.03	0.02
Tank 3	100	70	0.05	0.01
Number of Charging Tanks			3	
Charging Tanks	Capacity	Initial Oil Amount	Initial Comp. 1 (min, max)	Initial Comp. 2 (min, max)
Tank 1	100	30	0.0167(0.01,0.02)	0.0333(0.03,0.038)
Tank 2	100	50	0.03(0.025,0.035)	0.023(0.018,0.027)
Tank 3	100	30	0.0433(0.04,0.048)	0.0133(0.01,0.018)
Number of CDU			2	
Unit costs involved invessel operation			Unloading cost: 8, Sea waiting cost: 5	
Tank inventory unit costs			Storage tank: 0.05, Charging tank: 0.08	
Unit changeover cost for charged oil switch			50 (independent of sequence and CDU)	
Demand of mixed oils by CDUs			Oil mix 1:100,Oil mix 2:100,Oil mix 3:100	

Table 2.2: System Information for Example 2

Scheduling Horizon(# of unit time)		12	
Number of Vessel Arrivals		3	
	Arrival Time	Amount of Crude	Concentration of key component
Vessel 1	1	50	0.01
Vessel 2	5	50	0.085
Vessel 3	9	50	0.06
Number of Storage Tanks		3	
Storage Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min~max)
Tank 1	100	20	0.02 (0.01~0.03)
Tank 2	100	20	0.05 (0.04~0.06)
Tank 3	100	20	0.08 (0.07~0.09)
Number of Charging Tanks		3	
Charging Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min~max)
Tank 1	100	30	0.03 (0.025~0.035)
Tank 2	100	50	0.05 (0.045~0.065)
Tank 3	100	30	0.08 (0.075~0.085)
Number of CDU		2	
Unit costs involved in vessel operation		Unloading cost: 10, Sea waiting cost: 5	
Tank inventory unit costs		Storage tank: 0.05, Charging tank: 0.08	
Unit changeover cost for charged oil switch		50 (independent of sequence and CDU)	
Demand of mixed oils by CDUs		Oil mix 1:50,Oil mix 2:50,Oil mix 3:50	

Table 2.3: System Information for Example 3

industrial size problem (problem 4), the proposed formulation can be solved efficiently using 2351 nodes and 26.35 CPU seconds using GAMS/CPLEX 7.5. A Sun Ultra 60 is used for all computations in this thesis. The proposed formulation involves 485 continuous and 76 binary variables, compared with 991 and 105 of the discrete time formulation and requires 1298 constraints, almost half of the constraints required by discrete time formulation (2154). The resulting Gantt charts of crude-oil transferring between vessels, storage tanks, charging tanks and CDUs are shown in Figures 2.5 - 2.8. However, no comparison is provided with the schedules obtained by Lee et al. (1996) since this information was not provided.

The comparison of real and approximate values of inventory cost is provided in Table 2.6, which illustrates that the approximation of inventory levels in objective function is reasonable. Note that in order to provide a comparison with the discrete time formulation results appeared in the literature, constraints (2.6a) - (2.6d) are not considered for the proceeding results. The incorporation of the bilinear equations

Scheduling Horizon(# of unit time)			15
Number of Vessel Arrivals			3
	Arrival Time	Amount of Crude	Concentration of key component
Vessel 1	1	60	0.03
Vessel 2	6	60	0.05
Vessel 3	11	60	0.065
Number of Storage Tanks			6
Storage Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min~max)
Tank 1	10~90	60	0.031 (0.25~0.38)
Tank 2	10~110	10	0.03 (0.02~0.04)
Tank 3	10~110	50	0.05 (0.04~0.06)
Tank 4	10~110	40	0.065 (0.06~0.07)
Tank 5	10~90	30	0.075 (0.07~0.08)
Tank 6	10~90	60	0.075 (0.07~0.08)
Number of Charging Tanks			4
Charging Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min~max)
Tank 1	80	5	0.0317 (0.03~0.035)
Tank 2	80	30	0.0483 (0.043~0.05)
Tank 3	80	30	0.0633 (0.06~0.065)
Tank 4	80	30	0.075 (0.071~0.08)
Number of CDU			3
Unit costs involved in vessel operation			Unloading cost: 7, Sea waiting cost: 5
Tank inventory unit costs			Storage tank: 0.05, Charging tank: 0.06
Unit changeover cost for charged oil switch			30 (independent of sequence and CDU)
Demand of mixed oils by CDUs			Oil mix 1: 60, Oil mix 2: 60 Oil mix 3: 60, Oil mix 4: 60

Table 2.4: System Information for Example 4

Example	Variables	0−1 Variables	Constraints	Objective	Iterations
1	***139***	***24***	***352***	***236.75*** [1]	***845***
1(Lee et al.)	192	36	331	217.667	1,695
2	***341***	***56***	***906***	***390.70***	***15,992***
2(Lee et al.)	4,566	70	825	352.55	331,493 (21148)
3	***295***	***48***	***864***	***285.65***	***12,957***
3(Lee et al.)	581	84	1,222	296.56	>515,541 (60,663)
4	***485***	***76***	***1,298***	***429.34***	***86,594***
4(Lee et al.)	N/A (991)	N/A (105)	N/A (2,154)	N/A (420.99)	N/A (157,883)

[1] No direct comparison can be made since the objective obtained by the proposed methodology corresponds to an approximation of the reported value due to the continuous nature of the formulation

Table 2.5: Computational Results and Comparisons

(2.6a) - (2.6d) results in a mixed integer nonlinear formulation with the additional complexity of nonconvexity. Using CPLEX 7.5 as MIP solver and CONOPT as NLP solver, the solution of problem 4 requires approximately 6 hours, which is mainly contributed to MIP subproblems. Since the problem is nonconvex, the local solver used guarantees only local optimality of the solution. The computational results are presented in Table 2.7.

Example	***1***	***2***	***3***	***4***
Real Value of Inventory Cost	***91.80***	***145.56***	***103.44***	***229.73***
Approximated Inventory Cost	***88.00***	***130.70***	***104.40***	***266.59***

Table 2.6: Comparison of Real and Approximate Inventory Cost

Example	Objective	Major Iterations	Total CPU time	CPU time on MIP(sec)	CPU time on NLP(sec)
1	*225.00*	*3*	*0.96*	*0.94*	*0.02*
2	*325.80*	*3*	*1383.14*	*1383.01*	*0.13*
3	*282.73*	*3*	*26.56*	*26.46*	*0.10*
4	*341.10*	*3*	*21912.29*	*21912.10*	*0.19*

Table 2.7: Computational Results of MINLP Model

2.5 Summary

In this chapter, a continuous-time formulation is presented for the short-term scheduling of crude-oil unloading and charging with inventory control. It is shown that the proposed formulation results in fewer variables and constraints and can be efficiently solved using available MILP solvers. Several examples are provided to illustrate the capability of the proposed formulation comparing with existing approaches.

Nomenclature

Indices

i = storage tanks

j = charging tanks

k = key components

l = CDUs

n = event points

v = vessels

Sets

I = storage tanks

I_j = storage tanks which can transfer crude oil to charging tank j

I_v = storage tanks which can be fed by vessel v

J = charging tanks

J_i = charging tanks which can be fed by storage tank i

J_l = charging tanks which can charge CDU l

K = key components

L = CDUs

L_j = CDUs which can be charged by charging tank l

N = event points within the time horizon

V = vessels

V_i = vessels which can feed crude oil to storage tank i

Parameters

Cinvbi = inventory cost of charging tanks per volume per day

Cinvst = inventory cost of storage tanks per volume per day

Csea = sea waiting cost per day

Cset = changeover cost per time

Cunload = unloading cost per day

Db0(j,k) = initial concentration of component k in the crude mix of charging tank j

Dbmax(j,k) = maximum concentration of component k in the crude mix of charging tank j

Dbmin(j,k) = minimum concentration of component k in the crude mix of charging tank j

DM(j) = demand of crude mix from charging tank j

Ds0(i,k) = initial concentration of component k in the crude oil of storage tank i

Dsmax(i,k) = maximum concentration of component k in the crude oil of storage tank i

Dsmin(i,k) = minimum concentration of component k in the crude oil of storage tank i

Dv(v,k) = concentration of component k in the crude oil of vessel v

fmax = maximum volume flow rate

fmin = minimum volume flow rate

H = time horizon

NE = total number of event points

NST = total number of storage tanks

Tarr(v) = arrival time of vessel v

uv(v,i) = denote if vessel v can unload crude oil into storage tank i

Vb0(j) = initial volume of crude mix in charging tank j

Vbmax(j) = maximum capacity of charging tank j

Vmax = upper bound of volume of oil being transferred

Vmin = lower bound of volume of oil being transferred

Vs0(i) = initial volume of crude oil in storage tank i

Vsmax(i) = maximum capacity of storage tank i

Vv0(v) = initial volume of crude oil in vessel v

Variables

Bb(j,l,n) = volume of crude mix that charging tank j charges into CDU l at event point n

Bbb(j,l,k,n) = volume of component k that charging tank j charges into CDU l at event point n

Bs(i,j,n) = volume of crude oil that storage tank i transfers to charging tank j at event point n

Bss(i,j,k,n) = volume of component k that storage tank i transfer to charging tank j at event point n

Bv(v,i,n) = volume of crude oil that vessel v unloads into storage tank i at event point n

cost = operating cost

Tbf(j,l,n) = end time of charging tank j charging crude mix into CDU l at event point n

Tbs(j,l,n) = starting time of charging tank j charging crude mix into CDU l at event point n

Tsf(i,j,n) = end time of storage tank i transferring crude oil to charging tank j at event

point n

Tss(i,j,n) = starting time of storage tank i transferring crude oil to charging tank j at event point n

Tvf(v,i,n) = end time of vessel v unloading crude oil into storage tank i at event point n

Tvft(v,i,n) = time that vessel v finishes unloading crude oil into storage tank i

Tvs(v,i,n) = starting time of vessel v unloading crude oil into storage tank i at event point n

Tvst(v,i,n) = time that vessel v starts unloading crude oil into storage tank i

Vb(j,n) = volume of crude mix in charging tank j at event point n

Vbb(j,k,n) = volume of component k in charging tank j at event point n

Vs(i,n) = volume of crude oil in storage tank i at event point n

Vss(i,k,n) = volume of component k in storage tank i at event point n

Vv(v,n) = volume of crude oil in vessel v at event point n

x(v,i,n) = binary variables that assign the beginning of v unloading crude oil to i at event point n

y(i,j,n) = binary variables that assign the beginning of i transferring crude oil to j at event point n z(j,l,n) = binary variables that assign the beginning of j charging crude oil mix to l at event point n

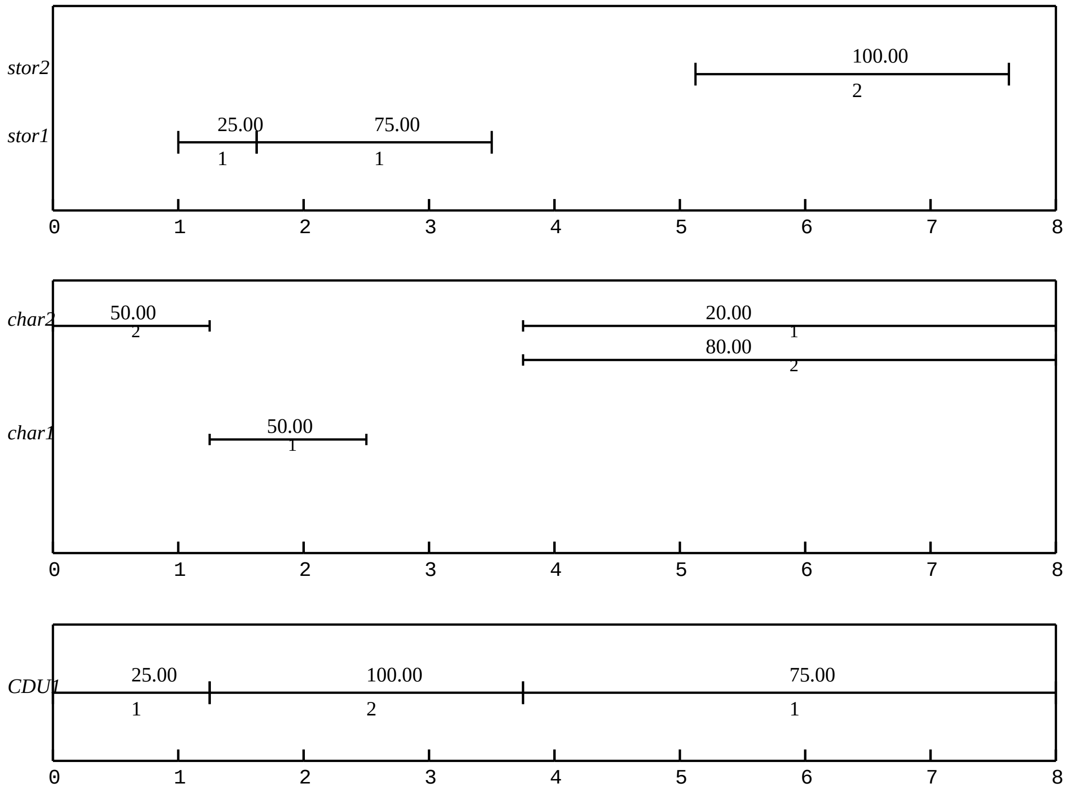


Figure 2.5: Gantt Chart of Operation Schedule for Example 1

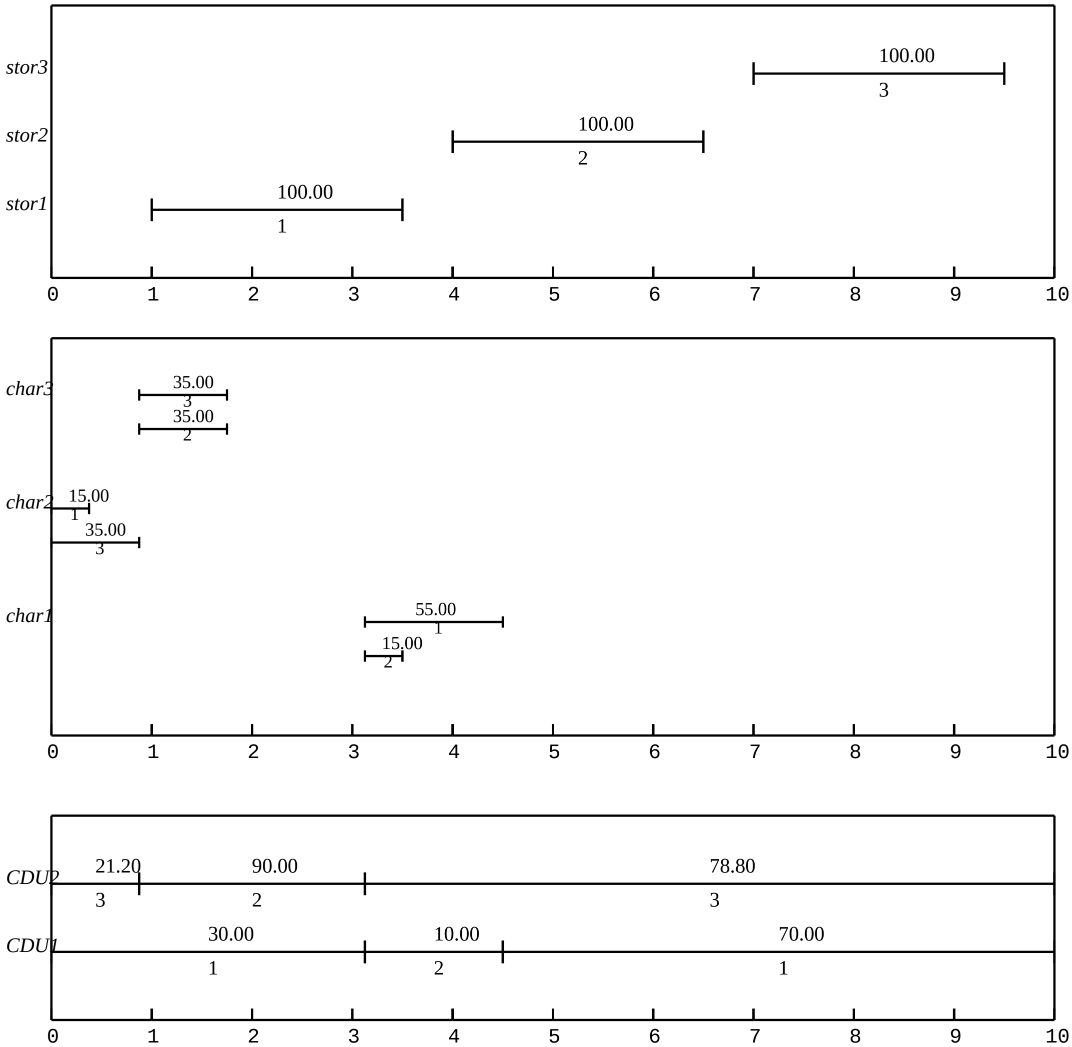


Figure 2.6: Gantt Chart of Operation Schedule for Example 2

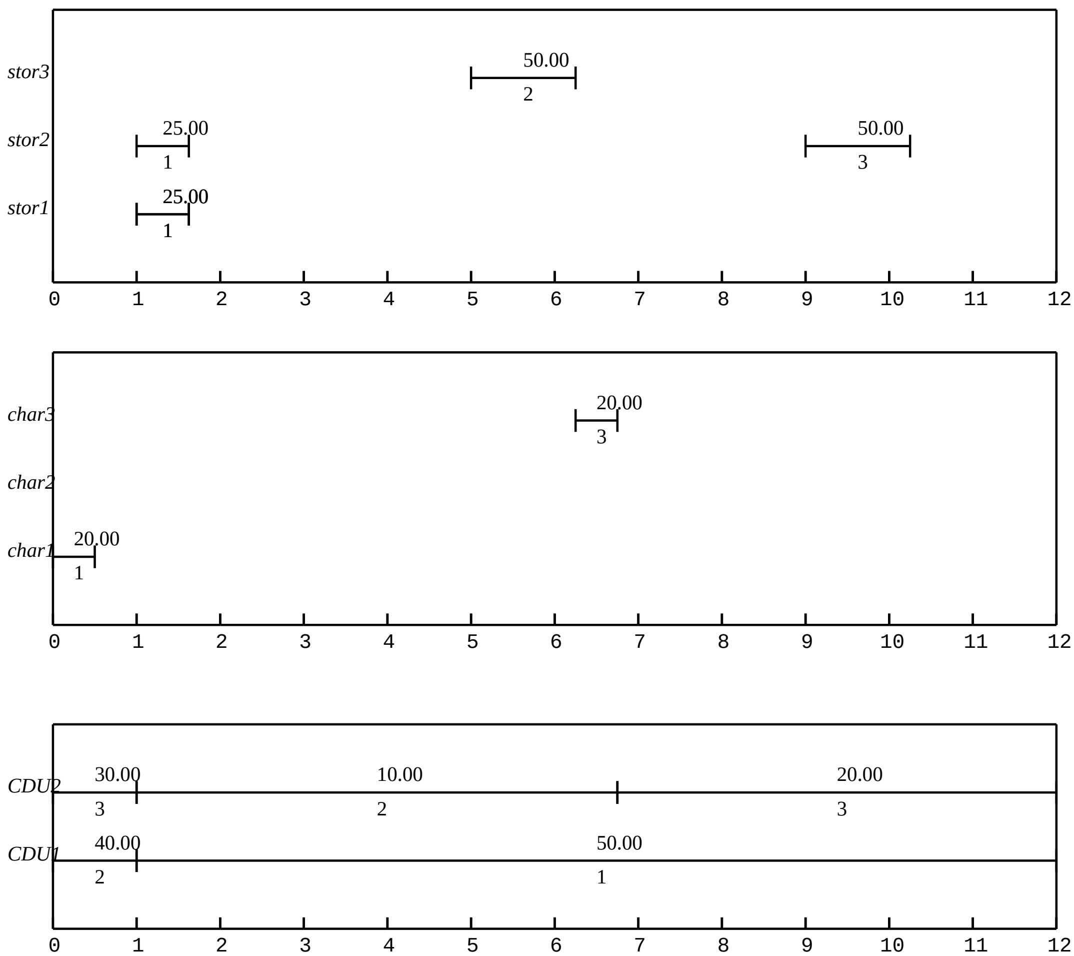


Figure 2.7: Gantt Chart of Operation Schedule for Example 3

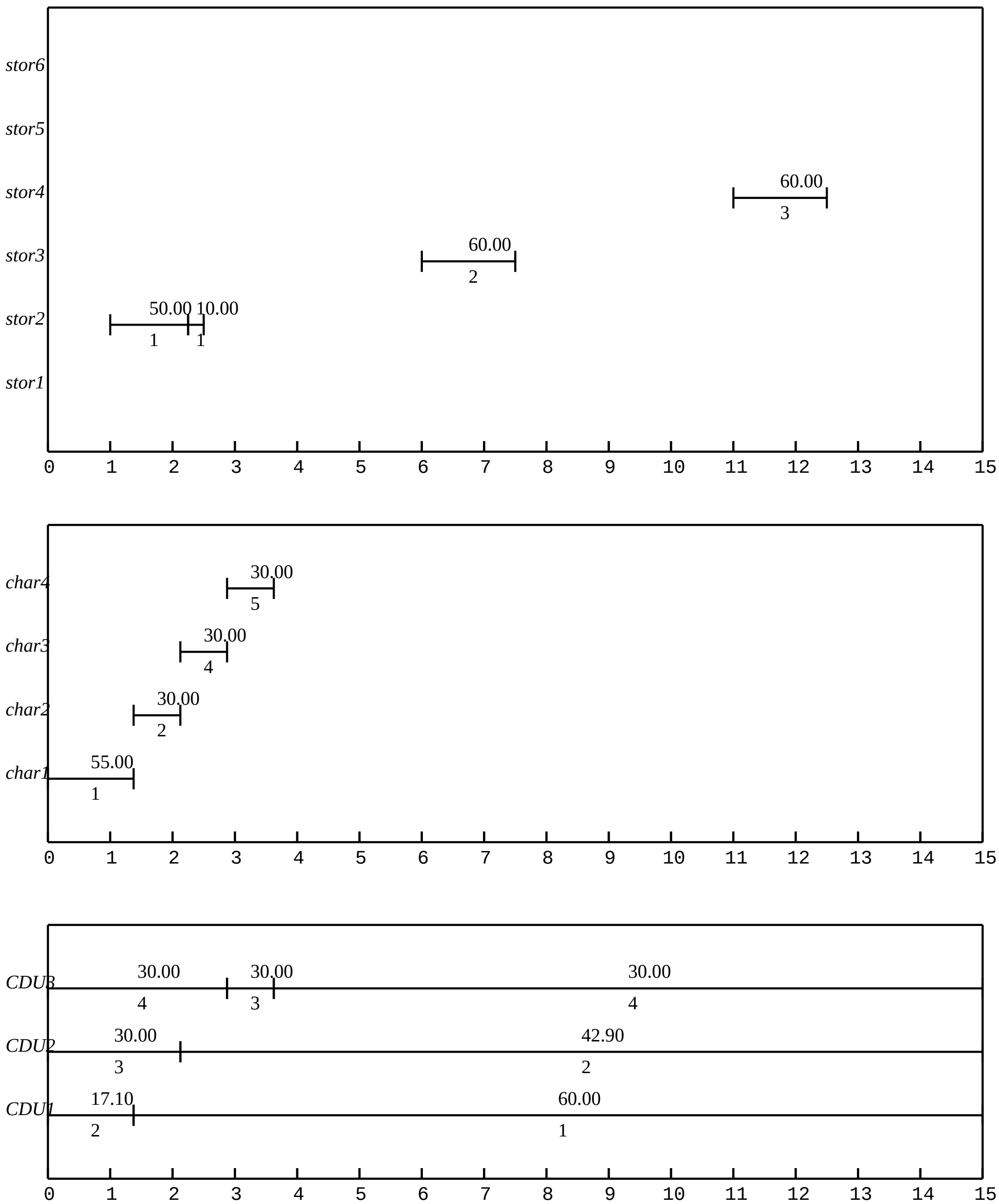


Figure 2.8: Gantt Chart of Operation Schedule for Example 4

Chapter 3

Production Stage

The second part of the overall oil-refinery system (Figure 1.1) considers the production unit scheduling. The scheduling problem is described in section 3.1 and modeled in section 3.2 based on a continous time formulation, which is applied to a realistic case study in section 3.3.

3.1 Problem Definition

This problem includes three processing stages: extraction, dewaxing and hydrofinishing. The key information available from the external sources are: a) yields between feed grades and product grades; b) minimum and maximum flowrates for each reaction; c) minimum and maximum inventory capacities for each storage tank; d) types of materials that can be stored in each storage tank; e) amount and type of final product required; f) the time horizon under consideration.

The objective is to determine the following variables: a) starting and end times of reactions taking place at each stage; b) amount and type of material being produced or consumed at each time in each production unit; c) amount and type of material being stored at each time in each tank, so as to maiximize the total profit during the time horizon.

The operating rules that have to be followed are: a) at most one material can be stored in one storage tank at one time interval; b) at most one reaction can take place in one production unit at one time interval.

3.2 Mathematical Formulation

It is assumed that perfect mixing is achieved at the production units and that the change-over time between different products in the storage tanks are neglegible. The proposed scheduling model is based on a continuous time representation and involves the following constraints:

Material Balance Constraints for Materials in Tank (j)

Constraints (3.1a) - (3.1h) state that the amount of materials in tank (j) 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). For raw material (s1), there is no amount produced from the previous stage while for final product (s4), there is no amount transferred to the next stage. At the first event point, the value of b(s,j,n-1) is given by the initial amount of (s) in the tank (initial(s,j)).

$$b1(s1,j,n) = b1(s1,j,n-1) - out1(s1,j,n), \forall s1 \in S1, j \in J_{s1}, n \in N, n \neq 1 \quad (3.1a)$$

$$b1(s1,j,n) = initial1(s1,j) - out1(s1,j,n), \forall s \in S1, j \in J_{s1}, n = 1 \quad (3.1b)$$

$$b2(s2,j,n) = b2(s2,j,n-1) - out2(s2,j,n) + in2(s2,j,n-1),$$
$$\forall s2 \in S2, j \in J_{s2}, n \in N, n \neq 1 \quad (3.1c)$$

$$b2(s2,j,n) = initial2(s2,j) - out2(s2,j,n), \forall s2 \in S2, j \in J_{s2}, n = 1 \quad (3.1d)$$

$$b3(s3,j,n) = b3(s3,j,n-1) - out3(s3,j,n) + in3(s3,j,n-1),$$
$$\forall s3 \in S3, j \in J_{s3}, n \in N, n \neq 1 \quad (3.1e)$$

$$b3(s3,j,n) = initial3(s3,j) - out3(s3,j,n), \forall s3 \in S3, j \in J_{s3}, n = 1 \quad (3.1f)$$

$$b4(s4,j,n) = b4(s4,j,n-1) + in4(s4,j,n-1), \forall s4 \in S4, j \in J_{s4}, n \in N, n \neq 1 \quad (3.1g)$$

$$b4(s4,j,n) = initial4(s4,j), \forall s4 \in S4, j \in J_{s4}, n = 1 \quad (3.1h)$$

Material Balance Constraints for Production Units

Constraints (3.2a) - (3.2d) express that for each production unit, the amount of product grade at event point (n) is equal to the that of feed grade at the same event point multiplying the yield. Since there is no one-to-one assignment between (s1) and (s2) at extraction stage, constraints (3.2a) and (3.2b) are active only if (s1) is consumed to produce (s2) at event point (n) (i.e., $uv1(s1,s2,n) = 1$).

$$\sum_{j \in J_{s1}} out1(s1,j,n) * yield1(s1,s2) \leq \sum_{j \in J_{s2}} in2(s2,j,n)) + U2 * (1 - uv1(s1,s2,n)),$$
$$\forall s1 \in S1, s2 \in S2_{s1}, n \in N \quad (3.2a)$$

$$\sum_{j \in J_{s1}} out1(s1,j,n) * yield1(s1,s2) \geq \sum_{j \in J_{s2}} in2(s2,j,n)) - U2 * (1 - uv1(s1,s2,n)),$$
$$\forall s1 \in S1, s2 \in S2_{s1}, n \in N \quad (3.2b)$$

$$\sum_{j \in J_{s2}} out2(s2,j,n) * yield2(s2,s3) = \sum_{j \in J_{s3}} in3(s3,j,n),$$
$$\forall s2 \in S2, s3 \in S3_{s2}, n \in N \quad (3.2c)$$

$$\sum_{j \in J_{s3}} out3(s3,j,n) * yield3(s3,s4) = \sum_{j \in J_{s4}} in4(s4,j,n),$$
$$\forall s3 \in S3, s4 \in S4_{s3}, n \in N \quad (3.2d)$$

Capacity Constraints

If there is material in tank (j) at event point (n), then constraints (3.3a) - (3.3d) correspond to the upper and lower bounds on the capacity (b(s,j,n)). If y(s,j,n) equals 1, then (b(s,j,n)) becomes zero.

$$lvol(j) * y1(s1,j,n) \leq b1(s1,j,n) \leq hvol(j) * y1(s1,j,n), \forall s1 \in S1, j \in J_{s1}, n \in N \quad (3.3a)$$

$$lvol(j) * y2(s2,j,n) \leq b2(s2,j,n) \leq hvol(j) * y2(s2,j,n), \forall s2 \in S2, j \in J_{s2}, n \in N \quad (3.3b)$$

$$lvol(j) * y3(s3,j,n) \leq b3(s3,j,n) \leq hvol(j) * y3(s3,j,n), \forall s3 \in S3, j \in J_{s3}, n \in N \quad (3.3c)$$

$$lvol(j) * y4(s4,j,n) \leq b4(s4,j,n) \leq hvol(j) * y4(s4,j,n), \forall s4 \in S4, j \in J_{s4}, n \in N \quad (3.3d)$$

Similarly, constraints (3.3e) - (3.3g) impose the upper and lower bounds on the amount of material being processed in the production units at event point (n). U1, U2 and U3 are obtained by summing up the maximum capacities of suitable tanks, while low is chosen as approximately $2 \sim 3$ times of normal flow rate.

$$low * uv1(s1, s2, n) \leq \sum_{j \in J_{s1}} out1(s1, j, n) \leq U1 * uv1(s1, s2, n), \forall s1 \in S1, n \in N \quad (3.3e)$$

$$low * uv2(s2, s3, n) \leq \sum_{j \in J_{s2}} out2(s2, j, n) \leq U2 * uv2(s2, s3, n), \forall s2 \in S2, n \in N \quad (3.3f)$$

$$low * uv3(s3, s4, n) \leq \sum_{j \in J_{s3}} out3(s3, j, n) \leq U3 * uv3(s3, s4, n), \forall s3 \in S3, n \in N \quad (3.3g)$$

Allocation Constraints

Constraint (3.4a) represents that at most one material can be stored in tank (j) at event point (n).

$$\sum_{s1} y1(s1, j, n) + \sum_{s2} y2(s2, j, n) + \sum_{s3} y3(s3, j, n) + \sum_{s4} y4(s4, j, n) \leq 1$$
$$\forall j \in J, s1 \in S1_j, s2 \in S2_j, s3 \in S3_j, s4 \in S4_j, n \in N \quad (3.4a)$$

According to constraints (3.4b) - (3.4d), at most one reaction can take place in one production unit at even point (n).

$$\sum_{s1} \sum_{s2} uv1(s1, s2, n) \leq 1, \forall s1 \in S1, s2 \in S2_{s1}, n \in N \quad (3.4b)$$

$$\sum_{s2} \sum_{s3} uv2(s2, s3, n) \leq 1, \forall s2 \in S2, s3 \in S3_{s2}, n \in N \quad (3.4c)$$

$$\sum_{s3} \sum_{s4} uv3(s3, s4, n) \leq 1, \forall s3 \in S3, s4 \in S4_{s3}, n \in N \quad (3.4d)$$

Mode Switch-over Constraints

If different raw materials (s1) and (s1$'$) are stored in the same tank consecutively, only certain switch-overs (switch(s1,s1$'$)=1) are allowed. Constraint (3.5) prevents all other disallowed switch-overs.

$$y1(s1, j, n) + y1(s1', j, n+1) \leq 1, \forall j \in J, s1 \in S1_j, s1' \in S1_j, switch(s1, s1') \neq 1,$$
$$n \in N, n \neq N \quad (3.5)$$

Demand Constraints

The initial amount of final product (s4) combined with the amount produced during the time horizon should meet the requirement (demand(s4)).

$$\sum_{j \in J_{s4}} \sum_{n} (in4(s4, j, n) + initial4(s4, j)) \geq demand(s4), \forall s4 \in S4, n \in N \quad (3.6)$$

Sequence Constrains for same reaction in the same unit

The following three sets of constraints state that one reaction starting at event point (n+1) should start after the end time of the same reaction in the same production unit which has started at event point n. Every reaction should start and end within the time horzion (H).

$$Ts1(s1, s2, n+1) \geq Te1(s1, s2, n) - H * (1 - uv1(s1, s2, n)),$$
$$\forall s1 \in S1, s2 \in S2_{s1}, n \in N, n \neq N \quad (3.7a)$$

$$Ts1(s1, s2, n+1) \geq Ts1(s1, s2, n), \forall s1 \in S1, s2 \in S2_{s1}, n \in N, n \neq N \quad (3.7b)$$

$$Te1(s1, s2, n+1) \geq Te1(s1, s2, n), \forall s1 \in S1, s2 \in S2_{s1}, n \in N, n \neq N \quad (3.7c)$$

$$Ts1(s1, s2, n) \leq H, \forall s1 \in S1, s2 \in S2_{s1}, n \in N \quad (3.7d)$$

$$Te1(s1, s2, n) \leq H, \forall s1 \in S1, s2 \in S2_{s1}, n \in N \quad (3.7e)$$

$$Ts2(s2, s3, n+1) \geq Te2(s2, s3, n) - H * (1 - uv2(s2, s3, n)),$$
$$\forall s2 \in S2, s3 \in S3_{s2}, n \in N, n \neq N \quad (3.8a)$$

$$Ts2(s2, s3, n+1) \geq Ts2(s2, s3, n), \forall s2 \in S2, s3 \in S3_{s2}, n \in N, n \neq N \quad (3.8b)$$

$$Te2(s2, s3, n+1) \geq Te2(s2, s3, n), \forall s2 \in S2, s3 \in S3_{s2}, n \in N, n \neq N \quad (3.8c)$$

$$Ts2(s2, s3, n) \leq H, \forall s2 \in S2, s3 \in S3_{s2}, n \in N \quad (3.8d)$$

$$Te2(s2, s3, n) \leq H, \forall s2 \in S2, s3 \in S3_{s2}, n \in N \quad (3.8e)$$

$$Ts3(s3, s4, n+1) \geq Te3(s3, s4, n) - H * (1 - uv3(s3, s4, n)),$$
$$\forall s3 \in S4, s4 \in S4_{s3}, n \in N, n \neq N \quad (3.9a)$$

$$Ts3(s3, s4, n+1) \geq Ts3(s3, s4, n), \forall s3 \in S4, s4 \in S4_{s3}, n \in N, n \neq N \quad (3.9b)$$

$$Te3(s3, s4, n+1) \geq Te3(s3, s4, n), \forall s3 \in S4, s4 \in S4_{s3}, n \in N, n \neq N \quad (3.9c)$$

$$Ts3(s3, s4, n) \leq H, \forall s3 \in S4, s4 \in S4_{s3}, n \in N \quad (3.9d)$$

$$Te3(s3, s4, n) \leq H, \forall s3 \in S4, s4 \in S4_{s3}, n \in N \quad (3.9e)$$

Sequence Constrains for different reactions in the same unit

If two reactions take place in the same production unit, then they should be perfomed at most consecutively.

$$Ts1(s1, s2, n+1) \geq Te1(s1', s2', n) - H * (1 - uv1(s1', s2', n))$$
$$\forall s1, s1' \in S1, s2 \in S2_{s1}, s2' \in S2'_{s1}, s2 \neq s2', n \in N, n \neq N \quad (3.10a)$$
$$Ts2(s2, s3, n+1) \geq Te2(s2', s3', n) - H * (1 - uv2(s2', s3', n))$$
$$\forall s2, s2' \in S2, s3 \in S3_{s2}, s3' \in S3'_{s2}, s3 \neq s3', n \in N, n \neq N \quad (3.10b)$$
$$Ts3(s3, s4, n+1) \geq Te3(s3', s4', n) - H * (1 - uv3(s3', s4', n))$$
$$\forall s3, s3' \in S3, s4 \in S4_{s3}, s4' \in S4'_{s3}, s4 \neq s4', n \in N, n \neq N \quad (3.10c)$$

Sequence Constrains for different reactions in different units

Constraints (3.11a) and (3.11b) are written for two consecutive stages that the product of the first stage is the feed of the second one. For example, in constraint (3.11a), if (s2) is produced at event point (n) (i.e., $uv1(s1, s2, n) = 1$), then the starting time of dewaxing stage where (s2) is consumed is later than finishing time of extraction stage.

$$Ts2(s2, s3, n+1) \geq Te1(s1, s2, n) - H * (1 - uv1(s1, s2, n))$$
$$\forall s1 \in S1_{s2}, s2 \in S2, s3 \in S3_{s2}, n \in N, n \neq N \qquad (3.11a)$$
$$Ts3(s3, s4, n+1) \geq Te2(s2, s3, n) - H * (1 - uv2(s2, s3, n))$$
$$\forall s2 \in S2_{s3}, s3 \in S3, s4 \in S4_{s3}, n \in N, n \neq N \qquad (3.11b)$$

Duration Constraints

Constraints (3.12a) - (3.12c) express the duration of each reaction should be between the amount of material being processed at that event point divided by the minimum and maximum flow rates.

$$\frac{\sum_{j \in J_{s1}} in2(s2, j, n)/yield1(s1, s2)}{maxrate1} \leq Te1(s1, s2, n) - Ts1(s1, s2, n)$$
$$\leq \frac{\sum_{j \in J_{s1}} in2(s2, j, n)/yield1(s1, s2)}{minrate1}, \forall s1 \in S1, s2 \in S2_{s1}, n \in N \qquad (3.12a)$$
$$\frac{\sum_{j \in J_{s2}} out2(s2, j, n)}{maxrate2} \leq Te2(s2, s3, n) - Ts2(s2, s3, n)$$
$$\leq \frac{\sum_{j \in J_{s2}} out2(s2, j, n)}{minrate2}, \forall s2 \in S2, s3 \in S3_{s2}, n \in N \qquad (3.12b)$$
$$\frac{\sum_{j \in J_{s3}} out3(s3, j, n)}{maxrate3} \leq Te3(s3, s4, n) - Ts3(s3, s4, n)$$
$$\leq \frac{\sum_{j \in J_{s3}} out3(s3, j, n)}{minrate3}, \forall s3 \in S3, s4 \in S4_{s3}, n \in N \qquad (3.12c)$$

Objective Function

The objective of this problem is to maximize the total profit over the time horizon, expressed as:

$$z = \sum_{s4} (\sum_{j \in J_{s4}} \sum_{n} (in4(s4, j, n) + initial4(s4, j))) * price(s4)), \forall s4 \in S4, n \in N \quad (3.13)$$

3.3 Case Studies

Scheduling Horizon (hrs)			168		
Feed Grade	Product Grade	Min Flowrate	Max Flowrate	Yield	Reactor
SP	SPLR	25	35	0.74	Extraction Unit
SP	SPHR	27	35	0.72	
LN	LNLR	24	38	0.71	
LN	LNHR	27	36	0.70	
SPLR	SPLD	30	72	0.72	Dewaxing Unit
SPHR	SPHD	28	70	0.74	
LNLR	LNLD	26	72	0.71	
LNHR	LNHD	28	70	0.72	
SPLD	SPLF	23	57	0.91	Hydro-finishing Unit
SPHD	SPHF	20	55	0.92	
LNLD	LNLF	23	58	0.93	
LNHD	LNHF	20	55	0.92	

Table 3.1: System Information for Problem 2

No. of Products	Continuous Variables	0-1 Variables	Constraints	Objective Value	Nodes	Iterations	CPU time (Sec)
4	6269	728	4388	2493.91	32	2417	2.58
6	18217	1668	11097	3855.86	134	27301	113.77
8	24307	2250	15095	4727.49	824	180086	1024.03
10	47255	4320	29220	5321.38	1747	734209	6431.16

Table 3.2: Computational Results for Problem 2

The case study considered here is based on realistic data provided by Honeywell Hi-Spec Solutions. The production unit scheduling problem consists of 5 raw materials, 3 processing stages and 10 products. Evaluation of the proposed formulation is performed by testing smaller scale instances of the problem involving the consideration of 4, 6, and 8 products. The detailed data are presented in Table 3.1. GAMS/CPLEX 7.0 is used for the solution of the resulting MILP formulation. The computational characteristics of the models are tabulated in Table 3.2. The resulting gantt charts of the example examined are shown in Figure 3.1. The numbers above the lines show the amounts of materials being processed in the reactors, while the numbers below the lines represent the feed grades of the reactions.

3.4 Summary

The scheduling problem of production stage is addressed in this chapter, based on a continuous-time formulation. The proposed methodology is applied to realistic case studies and can be efficiently solved using available MILP solvers.

Nomenclature

Indices

j = storage tanks

n = event points

s1 = raw materials

s2 = intermediates

s3 = intermediates

s4 = final products

Sets

J = storage tanks

J_{s1} = storage tanks which can store raw material s1

J_{s2} = storage tanks which can store intermediate s2

J_{s3} = storage tanks which can store intermediate s3

J_{s4} = storage tanks which can store final product s4

N = event points within the time horizon

S1 = raw materials

$S1_{s2}$ = raw materials which can produce material s2

S2 = intermediates

$S2_{s1}$ = materials which can be produced from raw material s1

$S2_{s3}$ = materials which can produce material s3

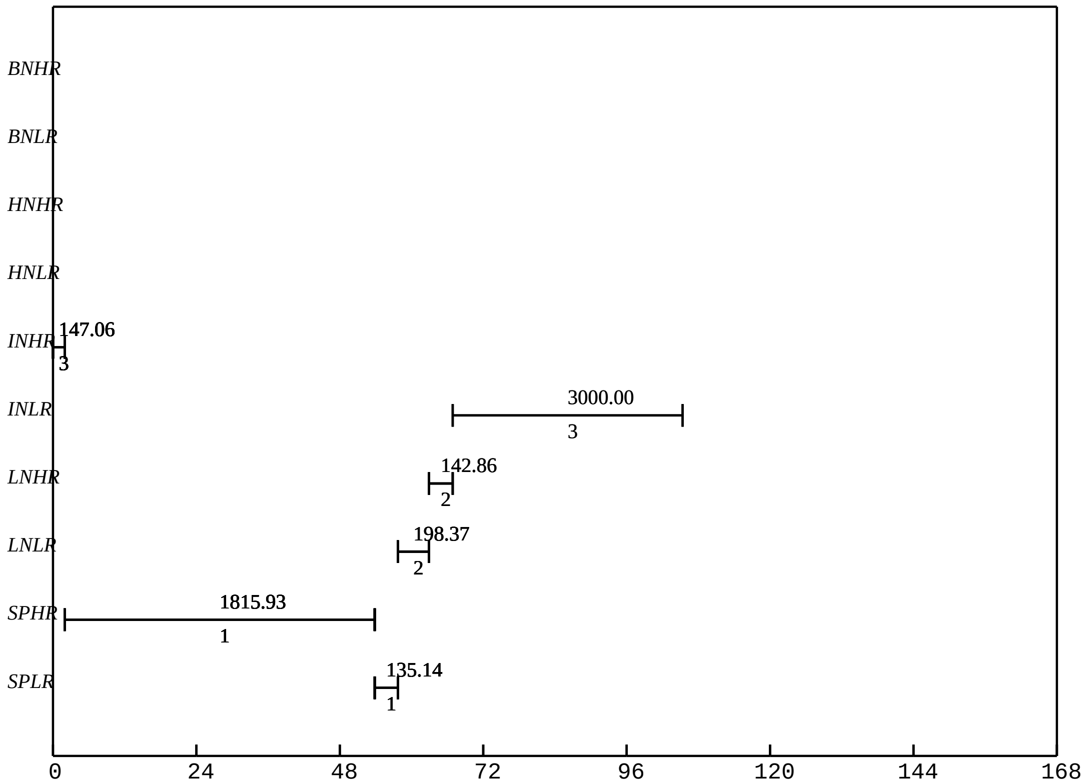
S3 = intermediates

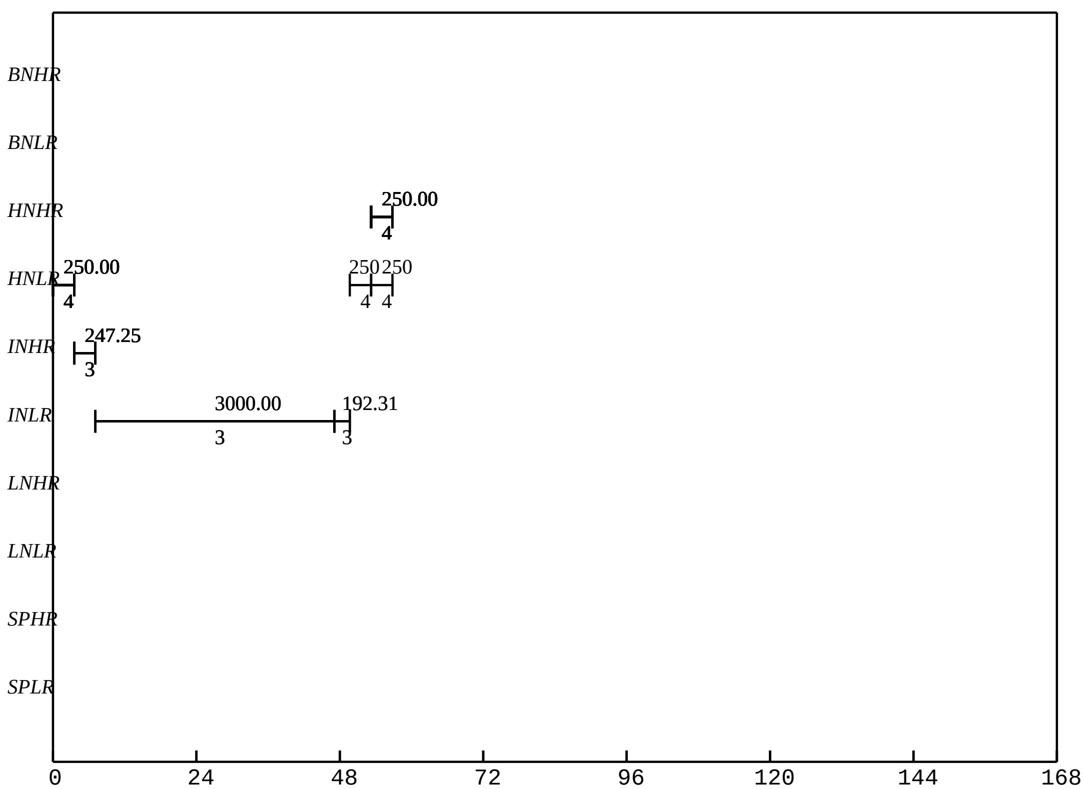
$S3_{s2}$ = materials which can be produced from material s2

$S3_{s4}$ = materials which can produce final product s4

S4 = final products

$S4_{s3}$ = final products which can be produced from material s3


BNHR
BNLR
HNHR
HNLR
INHR
147.06
3
INLR
3000.00
3
LNHR
142.86
2
LNLR
198.37
2
SPHR
1815.93
1
SPLR
135.14
1
0
24
48
72
96
120
144
168


BNHR
BNLR
HNHR
250.00
4
HNLR
250.00
4
250
4
250
4
INHR
247.25
3
INLR
3000.00
3
192.31
3
LNHR
LNLR
SPHR
SPLR
0
24
48
72
96
120
144
168

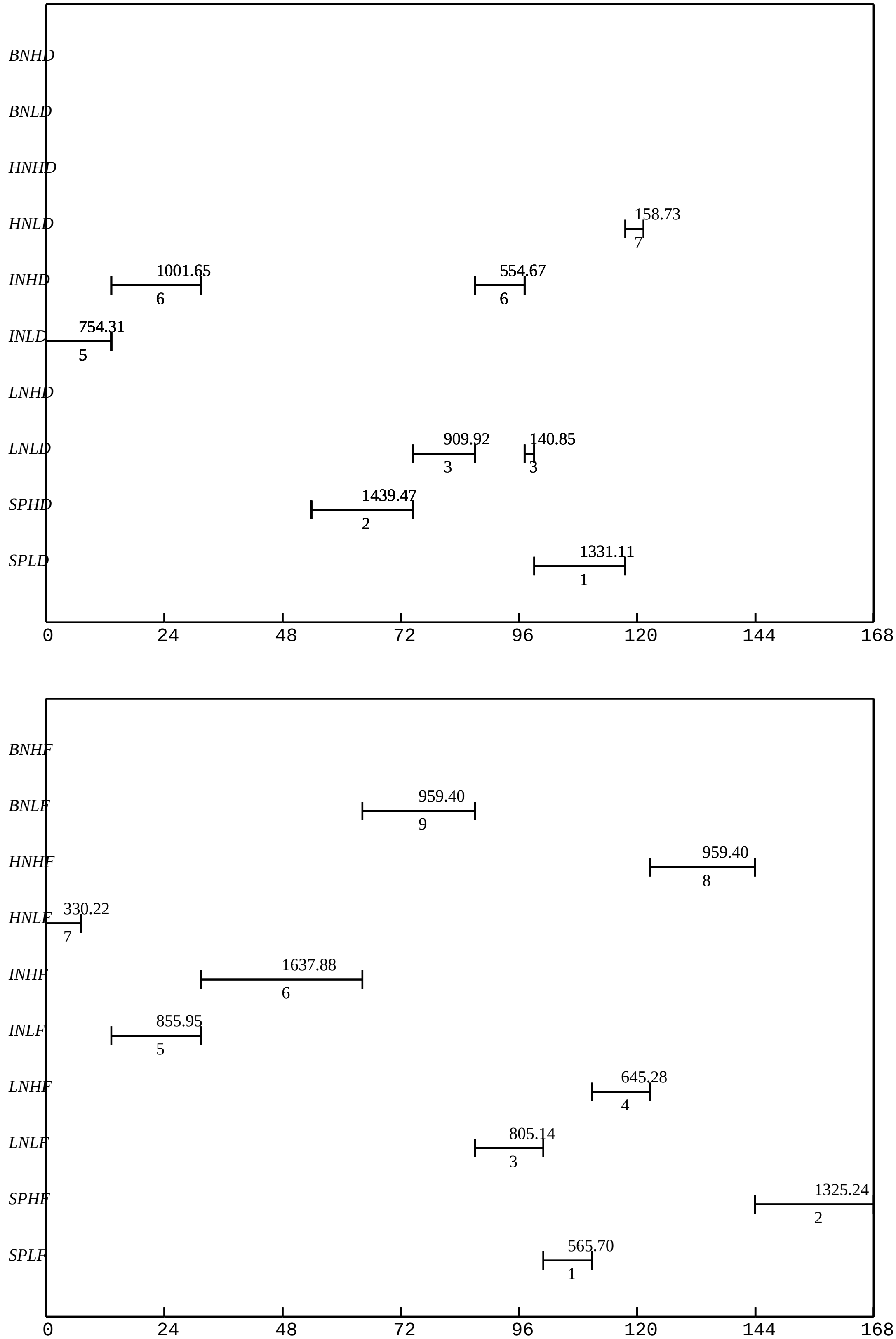


Figure 3.1: Gantt Charts for the Motivating Example

Chapter 4

Gasoline Blending and Distribution

4.1 Introduction

Gasoline blending is a crucial step in refinery operation as gasoline can yield 60-70% of a refinery's profit. The process involves mixing various stocks, which are the intermediate products from the refinery, along with some additives, such as antioxidants and corrosion inhibitors, to produce blends with certain qualities[29]. A variety of support systems have been developed to address planning and scheduling of blending operations. StarBlend[91], for example, which is developed by Texaco, uses a multi-period blending model written in GAMS that facilitates the incorporation of future requirements into current blending decisions. Glismann and Gruhn[41,42] proposed a mixed-integer linear model (MILP), which is based on a resource-task network representation, to solve the task of short-term scheduling of blending processes. The recipe optimization problem is then formulated as a non-linear program and the results are returned to the scheduling problem, so that an overall optimization can be achieved. A fuzzy linear formulation is applied to the blending facilities by Djukanovic et al.[30], in order to address the problem of uncertainty of input information within the fuel scheduling optimization. Singh et al.[105] addressed the problem of blending optimization for in-line blending for the case of stochastic disturbances in feedstock qualities. They presented a real-time optimization method that can provide significantly improved profitability. Deterministic global optimization methods, such as branch-and-bound, cutting plane algorithms were studied by Horst and Tuy[50]. Ryoo and Sahinidis [94] proposed a novel branch-and-bound based method for discrete/continuous global optimization, using the idea of range reduction tests of variables.

Following the spatial decomposition which distinguishes the receiving, producing and delivery end of the refinery, the problem considered in this chapter involves the optimal operation of gasoline blending, the transfer to productstock tanks and the delivering schedule to satisfy all the orders. This chapter is organized as follows. The problem of gasoline blending and distribution is described in section 4.2. Section 4.3 presents the mathematical formulation for the distribution problem. In section 4.4, the formulation is extended to incorporate the consideration of the blending stage in order to improve the flexibility of the production schedule. Computational results of the proposed formulation are presented in section 4.5 to illustrate the applicability and efficiency of the proposed approach to real case-studies provided by Honeywell Hi-Spec Solutions.

4.2 Problem Definition

The gasoline blending system consists of four pieces of equipment all linked together through various piping segments, flowmeters and valves. They are in order: componentstock tanks, blend header, productstock tanks and lifting ports. Components from the componentstock tanks are fed to the blend header according to the recipes. Thus, different products can be produced and then stored in their suitable productstock tanks. The final step is to lift those products during the specified time periods in order to satisfy all the orders. The key information available from the external sources are: a) times that orders can start and the due dates; b) amount and type of product required for each order; c) minimum and maximum inventory capacities for each productstock tank; d) types of products that can be stored in each productstock tank; e) recipe of each product which is assumed fixed to maintain model's linearity; f) minimum and maximum flowrate capacities for the blend header; g) the time horizon under consideration.

The objective is to determine the following variables:; a) starting and ending time of orders taking place in each productstock tank; b) amount and type of product being lifted for each order from tanks; c) starting and ending time of product being transferred from the blender to the tanks; d) amount and type of component being transferred from component tanks to the blender, so as to process all the orders in specific time periods.

The operating rules that have to be followed are: a) the blender can produce only one product at each time interval; b) if one product is transferred from blender to the productstock tank, the productstock tank cannot distribute the same product simultaneously and vice versa; c) at most one product can be stored in one productstock tank at each time interval; d) each order can be executed simultaneously from one or more productstock tanks.

4.3 Mathematical Formulation

It is assumed that perfect mixing is achieved at the blend header and that the change-over time between different products in the storage tanks are negligible. The proposed scheduling model is based on a continuous time representation and involves the following constraints.

Material Balance Constraints for Productstock Tank (j)

Constraint (4.1a) expresses that the amount of product (s) in tank (j) at event point (n+1) (Pst(s,j,n+1)) is equal to that at event point (n) adjusted by any amounts transferred from blender (Blnd(s,j,n)) or lifted at event point (n) ($\sum_{i \in I_s} lift(i,j,n)$). Constraint (4.1b) states that the amount of product (s) being lifted from tank (j) at the last event point (N) should not exceed the amount of product (s) stored in tank (j).

$$Pst(s,j,n+1) = Pst(s,j,n) + Blnd(s,j,n) - \sum_{i \in I_s} lift(i,j,n),$$
$$\forall s \in S, j \in J_s, n \in N, n \neq N \qquad (4.1a)$$

$$Pst(s,j,n) + Blnd(s,j,n) \geq \sum_{i \in I_s} lift(i,j,n), \forall s \in S, j \in J_s, n = N \qquad (4.1b)$$

Capacity Constraints

Constraint (4.2) imposes a volume capacity limitation of product (s) in tank (j) at event point (n).

$$Vmin(j) * y(s,j,n) \leq Pst(s,j,n) + Blnd(s,j,n) \leq Vmax(j) * y(s,j,n),$$
$$\forall s \in S, j \in J_s, n \in N \qquad (4.2)$$

Allocation Constraints

According to constraint (4.3a), uv(i,j,n) is equal to 1 if the amount of product being lifted from tank (j) for order (i) is not zero at event point (n), that is, $lift(i,j,n) \neq 0$; uv(i,j,n) equals 0, otherwise. U1 and U2 correspond to lower and upper bound on the amount of product lifted, respectively, and are chosen according to the smallest order and the maximum capacities of tanks.

$$U1 * uv(i,j,n) \leq lift(i,j,n) \leq U2 * uv(i,j,n), \forall i \in I, j \in J_i, n \in N \qquad (4.3a)$$

To avoid task splitting, constraints (4.3b) - (4.3d) state that order (i) should be processed only once if it is a small-size order and at most twice if it is a medium-size order. Otherwise, it can be processed at most three times. For different problems, U3 and U4 are chosen accordingly to define small-size and medium-size orders. Constraint (4.3e) expresses that for large-size orders which are defined as greater than or equal to U5, the minimum order splitting is 25Mbbl.

$$\sum_n \sum_{j\in J_i} uv(i,j,n) = 1, \forall \sum_s Prod\_ord(i,s) \leq U3, i \in I, n \in N \tag{4.3b}$$

$$\sum_n \sum_{j\in J_i} uv(i,j,n) \leq 2, \forall \sum_s Prod\_ord(i,s) \leq U4, i \in I, n \in N \tag{4.3c}$$

$$\sum_n \sum_{j\in J_i} uv(i,j,n) \leq 3, \forall i \in I, n \in N \tag{4.3d}$$

$$25 * uv(i,j,n) \leq lift(i,j,n), \forall \sum_s Prod\_ord(i,s) \geq U5, \quad i \in I, j \in J_i, n \in N \tag{4.3e}$$

Constraint (4.4) forces sv(s,j,n) to be equal to 1 when Blnd(s,j,n) is not zero, otherwise, sv(s,j,n) equals 0.

$$Vmin(j) * sv(s,j,n) \leq Blnd(s,j,n) \leq Vmax(j) * sv(s,j,n), \forall s \in S, j \in J_s, n \in N \tag{4.4}$$

Demand Constraints

Constraints (4.5a), (4.5b) state that order (i) can be processed at most once in one tank during the time horizon under consideration and that the amount of product being lifted from all the productstock tanks should be equal to the amount ordered ($\sum_s Prod\_ord(i,s)$).

$$\sum_n uv(i,j,n) \leq 1, \forall i \in I, j \in J_i, n \in N \tag{4.5a}$$

$$\sum_n \sum_{j\in J_i} lift(i,j,n) = \sum_s Prod\_ord(i,s), \forall s \in S, i \in I, n \in N \tag{4.5b}$$

Sequence Constraints

Constraints (4.6a) - (4.6c) state that order (i) starting in tank (j) at event point (n+1) should start after the ending time of the same order processed in the same tank which has started at event point (n). Constraints (4.6d) and (4.6e) express that order (i) should start and finish during the specific time period based on the order requirement. These constraints are relaxed if uv(i,j,n) is zero, which means the order

(i) is not executed at tank (j) at event point (n).

$$Ts(i,j,n+1) \geq Te(i,j,n) - H*(1-uv(i,j,n)), \forall\ i \in I_j, j \in J, n \in N, n \neq N \quad (4.6a)$$

$$Ts(i,j,n+1) \geq Ts(i,j,n), \forall i \in I_j, j \in J, n \in N, n \neq N \quad (4.6b)$$

$$Te(i,j,n+1) \geq Te(i,j,n), \forall i \in I_j, j \in J, n \in N, n \neq N \quad (4.6c)$$

$$Ts(i,j,n) \geq Prod\_srt(i) * uv(i,j,n), \forall i \in I_j, j \in J, n \in N \quad (4.6d)$$

$$Te(i,j,n) \leq Prod\_end(i) + H*(1-uv(i,j,n)), \forall i \in I_j, j \in J, n \in N \quad (4.6e)$$

Duration Constraints

If order (i) is processed in tank (j) at event point (n), that is, $uv(i,j,n) = 1$, then both ends of constraint (4.7a) are equal so that the duration is given by $lift(i,j,n)/l(i)$, where $l(i)$ is the lifting rate of order (i). If $uv(i,j,n) = 0$, then the duration is zero according to constraint (4.7b).

$$\frac{lift(i,j,n) - \sum_s Prod\_ord(i,s)*(1-uv(i,j,n))}{l(i)} \leq Te(i,j,n) - Ts(i,j,n) \leq \frac{lift(i,j,n)}{l(i)}, \forall i \in I_j, j \in J, n \in N \quad (4.7a)$$

$$Te(i,j,n) - Ts(i,j,n) \leq \frac{\sum_{s \in S_j} Prod\_ord(i,s)*uv(i,j,n)}{l(i)}, \forall i \in I_j, j \in J, n \in N \quad (4.7b)$$

4.4 Blending Stage Consideration

The consideration of the blending stage requires the incorporation of the following constraints:

Material Balance Constraints for the Blender

To avoid the introduction of bilinear terms in the mass balance equations and keep the model linear, the idea of component mixing used by[87] together with the assumption of constant production recipe is used. Based on these assumptions, constraint (4.8) is introduced to express that the required amount of component (k) in order to produce product (s) at event point (n) ($\sum_s(Recipe(s,k)*\sum_{j\in J_s} Blnd(s,j,n))$) should be equal to the total amount of component (k) being transferred from all the component tanks at that event point ($\sum_{l \in L_k} comp(k,l,n)$).

$$\sum_s (Recipe(s,k) * \sum_{j \in J_s} Blnd(s,j,n)) = \sum_{l \in L_k} comp(k,l,n) \forall s \in S, k \in K, n \in N \quad (4.8)$$

Note that the introduction of realistic nonlinear blending laws or even linear blending based on component properties would result in a MINLP with the additional complexity of the appearance of nonconvex bilinear terms, which correspond to a global optimization problem. As mentioned in the introduction, this problem alone is a subject of numerous publications in the global optimization literature. Thus, the assumption of constant recipe is made to overcome this burden and takle the integration of the product and delivery stages.

Material Balance Constraints for Component Tank (l)

The amount of component (k) in tank (l) at event point (n+1) (bc(k,l,n+1)) is equal to that at event point (n) (bc(k,l,n)) adjusted by any amounts transferred from separation units (cracking(k,l,n)) or delivered to the blender at event point (n) (comp(k,l,n)). This relation is expressed by constraint (4.9a). Constraint (4.9b) imposes the upper and the lower bounds on the flow rates of component (k) transferred from tank (l) to the blender.

$$bc(k,l,n+1) = bc(k,l,n) + cracking(k,l,n) - comp(k,l,n), \forall k \in K_l, n \in N \quad (4.9a)$$

$$flowmin * yv(k,l,n) \leq comp(k,l,n) \leq flowmax * yv(k,l,n),$$
$$\forall k \in K, l \in L_k, n \in N \quad (4.9b)$$

Allocation Constraints for Productstock Tank (j)

Constraint (4.10) states that product (s) cannot be transferred to productstock tank (j) and distributed at the same event point (n).

$$\sum_{s \in S_j} sv(s,j,n) + uv(i,j,n) \leq 1, \forall i \in I_j, j \in J, n \in N \quad (4.10)$$

Allocation Constraints for Blender

According to constraint (4.11a), xv(s,n) equals 1 if product (s) is produced and transferred to at least one tank at event point (n), whereas xv(s,n) equals 0 if product (s) is not transferred to any of the tanks at event point (n). Constraint (4.11b) expresses that only one product can be produced in the blender at the same event point (n).

$$sv(s,j,n) \leq xv(s,n) \leq \sum_{j \in J_s} sv(s,j,n), \forall s \in S, n \in N \tag{4.11a}$$

$$\sum_{s} xv(s,n) \leq 1, \forall s \in S, n \in N \tag{4.11b}$$

Sequence Constraints

Similar to constraints (4.6a) - (4.6c), constraints (4.12a) - (4.12c) state that product (s) should start being transferred to tank (j) at event point (n+1) after the ending time of the same product transferred to the same tank which has started at event point (n), whereas constraints (4.12d) and (4.12e) represent the requirement of all the transfers to happen within the time horizon (H).

$$Tbs(s,j,n+1) \geq Tbe(s,j,n) - H * (1 - sv(s,j,n)),$$
$$\forall s \in S_j, j \in J, n \in N, n \neq N \tag{4.12a}$$

$$Tbs(s,j,n+1) \geq Tbs(s,j,n), \forall s \in S_j, j \in J, n \in N, n \neq N \tag{4.12b}$$

$$Tbe(s,j,n+1) \geq Tbe(s,j,n), \forall s \in S_j, j \in J, n \in N, n \neq N \tag{4.12c}$$

$$Tbs(s,j,n) \leq H, \forall s \in S_j, j \in J, n \in N \tag{4.12d}$$

$$Tbe(s,j,n) \leq H, \forall s \in S_j, j \in J, n \in N \tag{4.12e}$$

If the blender provides product (s) for more than one productstock tanks at event point (n), then the starting and finishing times for all the tanks should be the same.

$$Tbs(s,j,n) + H*(1 - sv(s,j,n)) \geq Tbs(s,j',n) - H*(1 - sv(s,j',n)),$$
$$\forall s \in S, j \in J_s, j' \in J_s, j \neq j', n \in N \quad (4.13a)$$
$$Tbs(s,j,n) - H*(1 - sv(s,j,n)) \leq Tbs(s,j',n) + H*(1 - sv(s,j',n)),$$
$$\forall s \in S, j \in J_s, j' \in J_s, j \neq j', n \in N \quad (4.13b)$$

$$Tbe(s,j,n) + H*(1 - sv(s,j,n)) \geq Tbe(s,j',n) - H*(1 - sv(s,j',n)),$$
$$\forall s \in S, j \in J_s, j' \in J_s, j \neq j', n \in N \quad (4.13c)$$
$$Tbe(s,j,n) - H*(1 - sv(s,j,n)) \leq Tbe(s,j',n) + H*(1 - sv(s,j',n)),$$
$$\forall s \in S, j \in J_s, j' \in J_s, j \neq j', n \in N \quad (4.13d)$$

Constraints (4.14a) and (4.14b) express that product transfer and distribution should be performed consecutively in the same productstock tank (j).

$$Ts(i,j,n+1) \geq Tbe(s,j,n) - H*(1 - sv(s,j,n)), \forall i \in I_j, s \in S_j, j \in J,$$
$$n \in N, n \neq N \quad (4.14a)$$
$$Tbs(s,j,n+1) \geq Te(i,j,n) - H*(1 - uv(i,j,n)), \forall i \in I_j, s \in S_j, j \in J,$$
$$n \in N, n \neq N \quad (4.14b)$$

According to constraint (4.15), two different products (s) and (s$^{'}$) being transferred to the same or different productstock tanks have to be transferred consecutively according to the allocation constraint for the blender.

$$Tbs(s,j,n+1) \geq Tbe(s',j',n) - H*(1 - sv(s',j',n)), \forall s \in S_j, s' \in S_j, s \neq s',$$
$$j \in J, j' \in J, n \in N, n \neq N \quad (4.15)$$

Duration Constraints

The minimum run length of 6 hours is imposed to the blender by constraint (4.16a)

$$\sum_{j \in J_s} Blnd(s,j,n) \geq 6 * Bflow, \forall s \in S, n \in N \tag{4.16a}$$

Constraint (4.16b) defines the duration of product (s) being transferred to the tanks at event point (n) as the difference between the ending time (Tbe(s,j,n)) and starting time (Tbs(s,j,n)), if it takes place in tank (j). Constraint (4.16c) expresses that the duration of transferring product (s) from the blender to tank (j) corresponds to the amount of product (s) being transferred divided by the flow rate. The purpose of having an artificial variable (arti(s,n)) is to find a feasible solution in case a larger flow rate is required.

$$(Tbe(s,j,n) - Tbs(s,j,n)) - H * (1 - sv(s,j,n)) \leq$$
$$duration(s,n) \leq (Tbe(s,j,n) - Tbs(s,j,n)) + H * (1 - sv(s,j,n)),$$
$$\forall s \in S_j, j \in J, n \in N \tag{4.16b}$$
$$duration(s,n) = \frac{\sum_{s \in S_j} Blnd(s,j,n)}{Bflow} - arti(s,n),$$
$$\forall s \in S_j, j \in J, n \in N \tag{4.16c}$$

Objective Function

The objective of the scheduling problem is to minimize the sum of artificial variables in the duration constraints of blender so as to determine a feasible solution with a flow rate as close to (Bflow) as possible. The formulation however is general to accommodate different objective functions targeting the optimization of production. However, in most realistic cases the objective of this stage of refinery operation is to satisfy all the orders without any delays.

$$objective = \sum_s \sum_n arti(s,n), \forall s \in S, n \in N \tag{4.17}$$

Order	o1	o2	o3	o4	o5	o6	o7	o8	o9	o10
Product & amount	N4 11	W4 3	W4 3	N4 11	W4 3	N4 11	W4 3	N4 132	W4 3	N5 175
Time that can start	0	0	24	24	48	48	96	118	144	150.5
Due date	24	24	48	48	72	72	120	190	168	185.5
Lifting rate	50	50	50	50	50	50	50	8	50	5
Time horizon	192									

Productstock tank	pt1	pt2	pt3	pt4	pt5	pt6	pt7	pt8	pt9	pt10	pt11
Products that can be stored	E4 W4	E4 W4	E4 W4	E4 N5 W4	E4 W4	E4 W4	N4 N5	N4 N5	N4 N5	N4 N5	N4 N5
Initial product & amount	E4 90.20	—	W4 14.08	N5 87.51	W4 28.49	W4 57.59	N4 13.79	N4 12.36	N5 23.96	N4 85.11	N4 12.36
Max capacity	92	92	94	91	92	84	94	92	92	91	82
Min capacity	0.92	0.92	0.94	0.91	0.92	0.84	0.94	092	0.92	0.91	0.82

Table 4.1: Distribution Data for Example with 10 Orders

Component		A	C7	C6	M	C4	C5	CR	AR	CG
Tanks that can be stored in		ct10	ct9	ct8	ct15,52 ct53,54	ct51	ct57,58 ct60	ct4 ct13	ct11 ct55	ct7,12,17 ct56,59
Recipe of products	N4	0	0.0767	0	0	0.14	0.2742	0.4018	0	0.1073
	N5	0	0	0.0419	0	0.0121	0.5178	0	0.0443	0.384
	E4	0	0	0	0	0.2729	0	0.3897	0	0.3078
	W4	0.6527	0	0	0	0.1591	0	0.1882	0	0
Amount of component and tanks that it is initially stored		26.46 ct10	67.90 ct9	59.44 ct8	7.30 ct15 5.75 ct52 3.10 ct53 28.29 ct54	0.59 ct51	0.29 ct57 8.90 ct58 1.64 ct60	27.38 ct4 19.35 ct13	25.63 ct11 13.84 ct55	34.58 ct7 49.34 ct51 53.41 ct56 4.25 ct59
Blending rate		50								

Table 4.2: Blending Data for Example with 10 Orders

Orders	Continuous Variables	0−1 Variables	Constraints	1st Integer Solution				2nd Integer Solution			Optimal Solution		
				Nodes	Iterations	CPU time (sec)	Obj. Value	Nodes	Iterations	Obj. Value	Nodes	Iterations	CPU time (sec)
10	1706	420	7130	21	1495	6.15	0	N/A	N/A	N/A	21	1495	6.15
16	4205	1032	18737	20	3614	29.03	0	N/A	N/A	N/A	20	3614	29.03
23	5974	1470	26746	40	13474	210.24	0	N/A	N/A	N/A	40	13474	210.24
30	9056	2232	40793	80	24906	627.13	0	N/A	N/A	N/A	80	24906	627.13
37	13955	3444	63308	828	177746	4081.49	4.934	1338	244258	0.793	1353	246176	5016.51
45	25454	6289	116452	361	194954	7406.48	6.645	4138	838195	5.094	4280	874051	20351.18

Table 4.3: Computational Results for the Blending and Distribution System

4.5 Case Studies

The case study considered here is based on realistic data provided by Honeywell Hi-Spec Solutions. The distribution problem consists of 45 orders of 4 different products that are stored in 11 productstock tanks. The incorporation of the blending stage adds the consideration of 9 components and 20 component tanks. Smaller-scale instances of the problem are constructed to test the proposed formulation involving the consideration of 10, 16, 23, 30, and 37 orders. The detailed data for the case of 10 orders are presented in Tables 4.1 and 4.2. GAMS/CPLEX 7.0 is used for the solution of the resulting MILP formulation. The computational characteristics of the models are tabulated in Table 4.3. The optimal solution with zero integrality gap as well as the first and second integer solutions are shown. Note that since the objective corresponds to the summation of artificial variables used to relax the flowrate constraints, if a solution has none-zero objective this indicates that one of these constraints has been violated at the cost of the objective function. Then depending on the temporal location and severity of violation the solution may be implemented, if global optimality is indeed expensive in the context of hierarchical scheduling. Given that scheduling is traditionally repeated at some regular interval to mitigate uncertainty, infeasibilities which occur some time out in the future may be aberrants of unreliable forecasted demands or simple effects of measurement error concerning the current inventory positions. Thus when used with discretion, it is possible and effective to take a preempted optimized solution and execute it provided it is done so as part of a scheduling cycle. For the case study examined however as shown in Table 4.3, even the full scale problem involving 45 orders converged to a feasible solution requiring 4280 nodes in approximately 5 hours CPU time which is a reasonable time for the solution of the integrated scheduling of blending and distribution problem with a time horizon of 8 days. The resulting Gantt charts of the six cases examined are shown in Figures 4.1 - 4.6. Compared to the commonly used Gantt-chart for scheduling purpose the difference here is that the number below the line corresponds to the order number, whereas the number above to the amount of product lifted from this particular tank. Note that different orders can be performed in the same tank at the same time as shown in Figures 4.2 - 4.6.

4.6 Summary and Future Work

In this chapter, a continuous-time formulation is presented for the short-term scheduling of gasoline blending and distribution system. It is shown that the resulting model can be solved efficiently even for realistic large-scale problems. The main advantage of the proposed approach is the full utilization of the time continuity. This results in smaller models in terms of variables and constraints since only the real events have to be modeled. In contrary, discrete time formulations which are commonly used for refinery operations result in excessive number of variables and constraints due to unnecessary time discretization.

In chapters 2, 3 and 4, comprehensive mathematical programming models are developed for the efficient scheduling of oil-refinery operations. However, in order to solve the short-term scheduling for refinery operations, it requires the integration of all three different problems. This integration, which can be achieved by using the output of one stage as the input of its following stage, will result in a large-scale problem that requires the utilization of decomposition approaches. One of the approaches that we propose to exploit is the idea of heuristic based Lagrangean decomposition iterative method.

4.6.1 Lagrangean Relaxation

The general integer programming problem can be written as:

$$\begin{aligned} maximize \quad & Z = cx \\ subject \quad to \quad & Ax \leq b \\ & Dx \leq e \\ & x \geq 0,\ x \ \ integer \end{aligned} \tag{4.18}$$

Assume that the constraints of (4.18) have been partitioned into two sets $Ax \leq b$ and $Dx \leq e$. The Lagrangean Relaxation of this proble m with repect to the first set of constraints has the following form:

$$\begin{aligned} maximize \quad & Z_D(u) = cx + u(b - Ax) \\ & Dx \leq e \\ & x \geq 0,\ x \ \ integer \end{aligned} \tag{4.19}$$

where u is a vector of Lagrange multipliers. By assuming an optimal solution x^* to problem (4.18), it is easy to observe that:

$$Z_D(u) \geq cx^* + u(b - Ax^*) = Z$$

Therefore, $Z_D(u)$ provides an upper bound on the original problem.

4.6.2 Lagrangean Decomposition

Guignard and Kim[44] proposed the Lagrangean decomposition (LD) scheme and showed that the decomposition of an integer programming problem into two sub-problems which share the constraints of the original problem can yield good bounds. Consider the following mathematical programming problem:

$$\begin{aligned} \textit{maximize} \quad & f^T x + d^T(y^1 + y^2) \\ \textit{subject to} \quad & Ax + B^1 y^1 \le b^1 \\ & Cx + B^2 y^2 \le b^2 \\ & x \in X,\ y^1, y^2 \in (0,1) \end{aligned} \tag{4.20}$$

A set of the constraints are relaxed such that the remaining model (4.21) is decomposable into two or more sub-problems that are easier to solve.

$$\begin{aligned} \textit{maximize} \quad & f^T x + d^T(y^1 + y^2) + u(z - x) \\ \textit{subject to} \quad & Ax + B^1 y^1 \le b^1 \\ & Cz + B^2 y^2 \le b^2 \\ & x, z \in X,\ y^1, y^2 \in (0,1) \end{aligned} \tag{4.21}$$

Then problem (4.21) can be decomposed into sub-problem (4.22) and (4.23), that can be solved independently.

$$\begin{aligned} \textit{maximize} \quad & f^T x + d^T y^1 - ux \\ \textit{subject to} \quad & Ax + B^1 y^1 \le b^1 \\ & x \in X,\ y^1 \in (0,1) \end{aligned} \tag{4.22}$$

$$\begin{aligned} \textit{maximize} \quad & d^T y^2 + uz \\ \textit{subject to} \quad & Cz + B^2 y^2 \le b^2 \\ & z \in X,\ y^2 \in (0,1) \end{aligned} \tag{4.23}$$

Note that if Z^1 and Z^2 are optimal solutions of sub-problem (4.22) and (4.23), then $Z_u^{UB} = Z^1 + Z^2$ is an upper bound of problem (4.20) for any u.

4.6.3 Iterative Solution Framework

Based on a set of heuristic based techniques to provide a valid lower bound and LR/LD, Wu and Ierapetritou[112] proposed an algorithmic procedure for the eff icient solution of large-scale scheduling problems as illustrated in Figure 4.7.

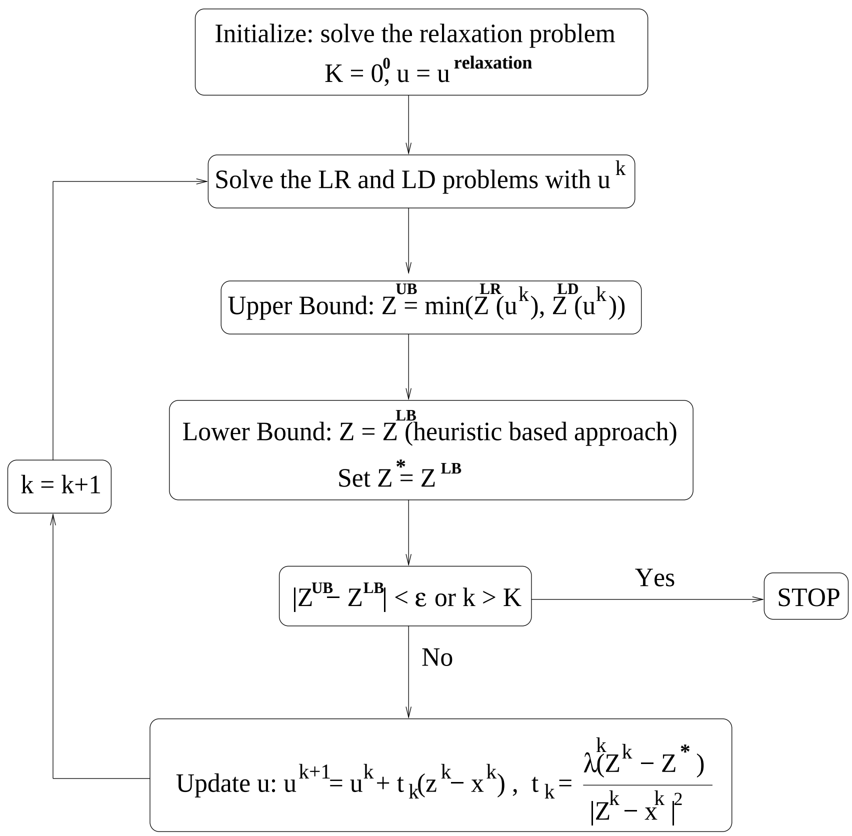


Figure 4.7: Algorithmic Procedure for Solving Large-scale Scheduling Problem

In their proposed solution framework, sub-gradient method is used to find the tightest bound $Z_D(u)$, which involves calculating the step size $t_k = \frac{\lambda^k(Z_D(u^k)-Z^*)}{|b-Ax^k|^2}$ and updating the Lagrangean multiplier u according to $u^{k+1} = u^k + t_k(Z^k - x^k)$. Various heuristic based approaches, such as time-based decomposition, required production method, and resource-based decomposition are utilized to provide a lower bound, while an upper bound of the scheduling problem is obtained from LR/LD. This procedure terminates when the gap between the upper and lower bound is less than a specified tolerance or if a certain number of iterations is reached. The efficiency of this approach has been illustrated through a number of examples.

Therefore, this heuristic based Lagrangian decomposition iterative methodology can be employed for the solution of the integrated model, which describes the scheduling problem of refinery operations including all three sub-problems. The models of the individual problems will be connected with the output of one stage considered as the input of its following stage.

Nomenclature

Indices

i = orders

j = productstock tanks

k = components

l = component tanks

n = event points

s = products

Sets

I = orders

I_j = orders which can be performed in productstock tank j

I_s = orders which order product s

J = productstock tanks

J_i = productstock tanks which are suitable for performing order i

J_s = productstock tanks which can store product s

K = components

K_l = components which can be stored in componentstock tank l

L = componentstock tanks

L_k = componentstock tanks which can store component k

N = event points within the time horizon

S = products

S_j = products which can be stored in productstock tank j

Parameters

Bflow = flow rate of product being produced and transferred to productstock tanks

flowmin = minimum flow rate of component tanks

flowmax = maximum flow rate of component tanks

H = time horizon

l(i) = lifting rate of order i

Prod_srt(i) = time by which order i can start

Prod_end(i) = time by which order i is due

Recipe(s,k) = the proportion of component k to in product s

U1 = lower bound on the amount of product lifted

U2 = upper bound on the amount of product lifted

U3 = upper bound of a small-size order

U4 = upper bound of a medium-size order

U5 = lower bound of a large-size order

Vmax(j) = maximum capacity of productstock tank j

Vmin(j) = minimum amount of product stored in tank j if tank j is utilized

Variables

bc(k,l,n) = amount of component k in comp. tank l at event point n

Blnd(s,j,n) = amount of product s being transferred from blender to tank j at event point n

comp(k,l,n) = amount of component k being transferred to the blender at event point n

cracking(k,l,n) = amount of component k being transferred from separation units to comp. tank l at event point n

lift(i,j,n) = amount of product being lifted for order i from tank j at event point n

Pst(s,j,n) = amount of product s in tank j at event point n before new product being transferred from blender

sv(s,j,n) = binary variables that assign product s being produced and transferred to tank j at event point n

Ts(i,j,n) = starting time of order i in tank j at event point n

Te(i,j,n) = ending time of order i in tank j while it starts at event point n

Tbs(s,j,n) = starting time of product s being produced and transferred to productstock tank j at event point n

Tbf(s,j,n) = ending time of product s being produced and transferred to productstock tank j at event point n

uv(i,j,n) = binary variables that assign the beginning of order i in tank j at event point n

xv(s,n) = 0-1 continuous variables that assign product s being produced at event point n

y(s,j,n) = binary variables that assign product s being stored in tank j at event point n

yv(k,l,n) = binary variables that assign component k being extracted from comp tank l at event point n

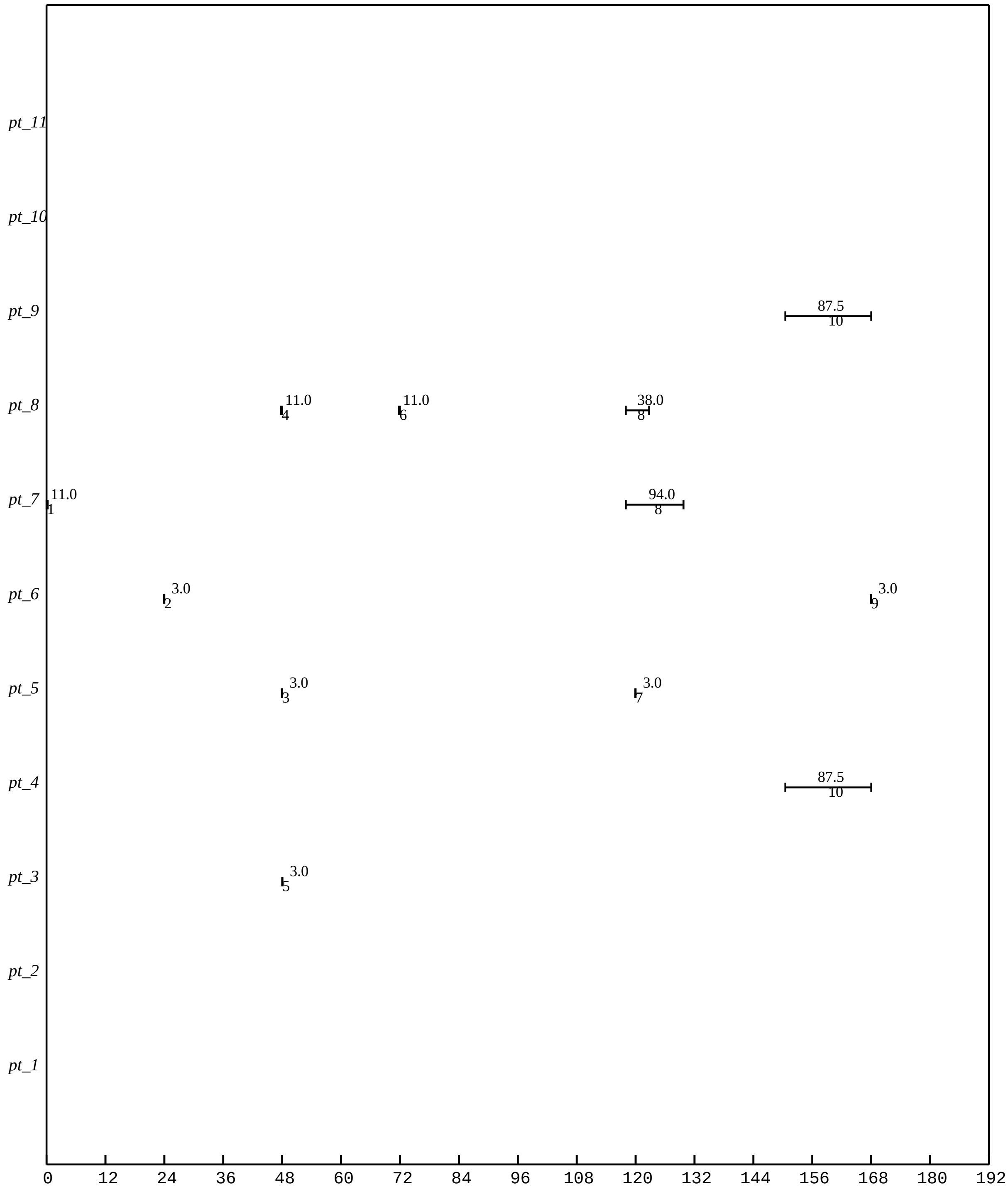


Figure 4.1: Gantt Chart for the Example with 10 Orders

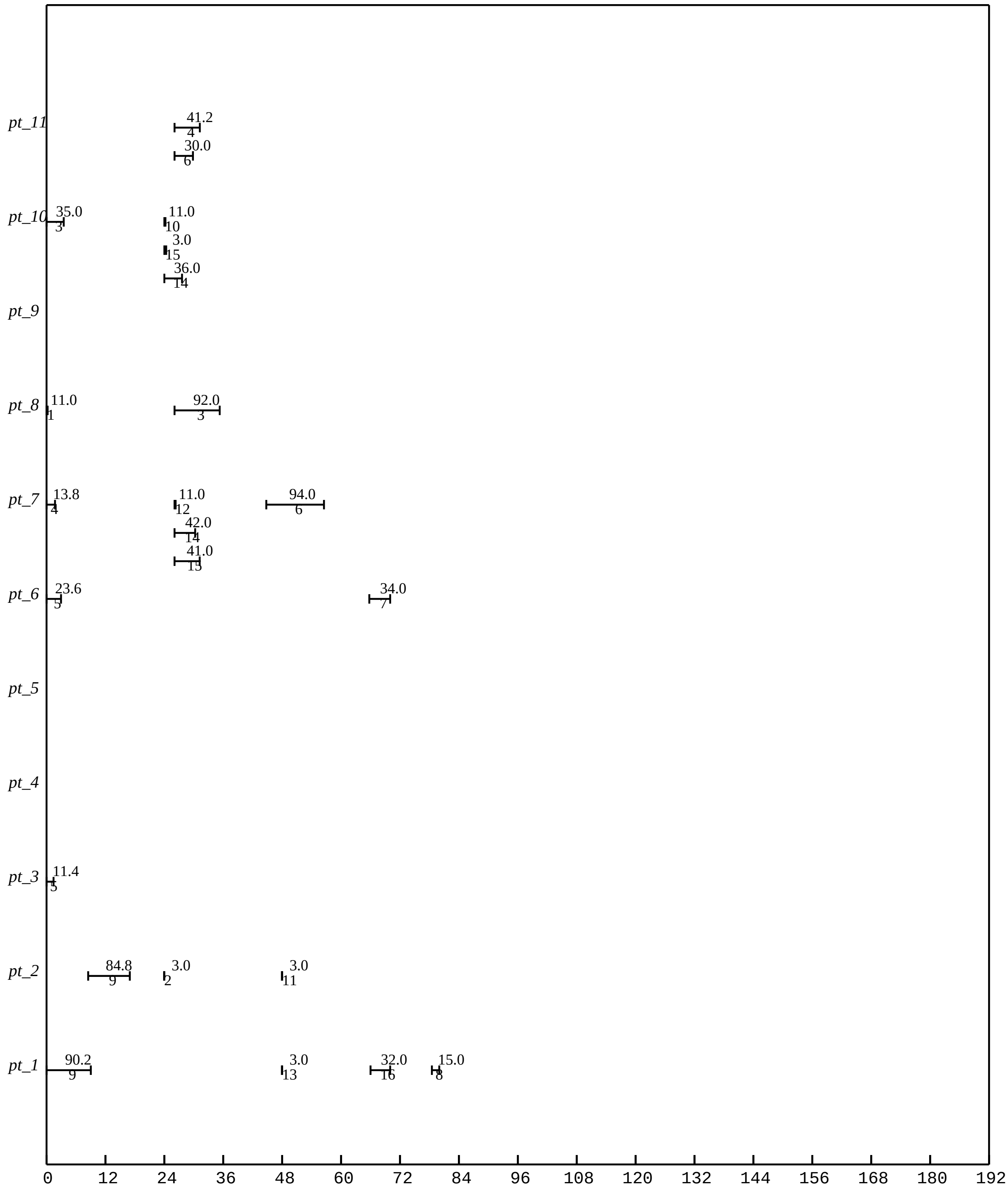


Figure 4.2: Gantt Chart for the Example with 16 Orders

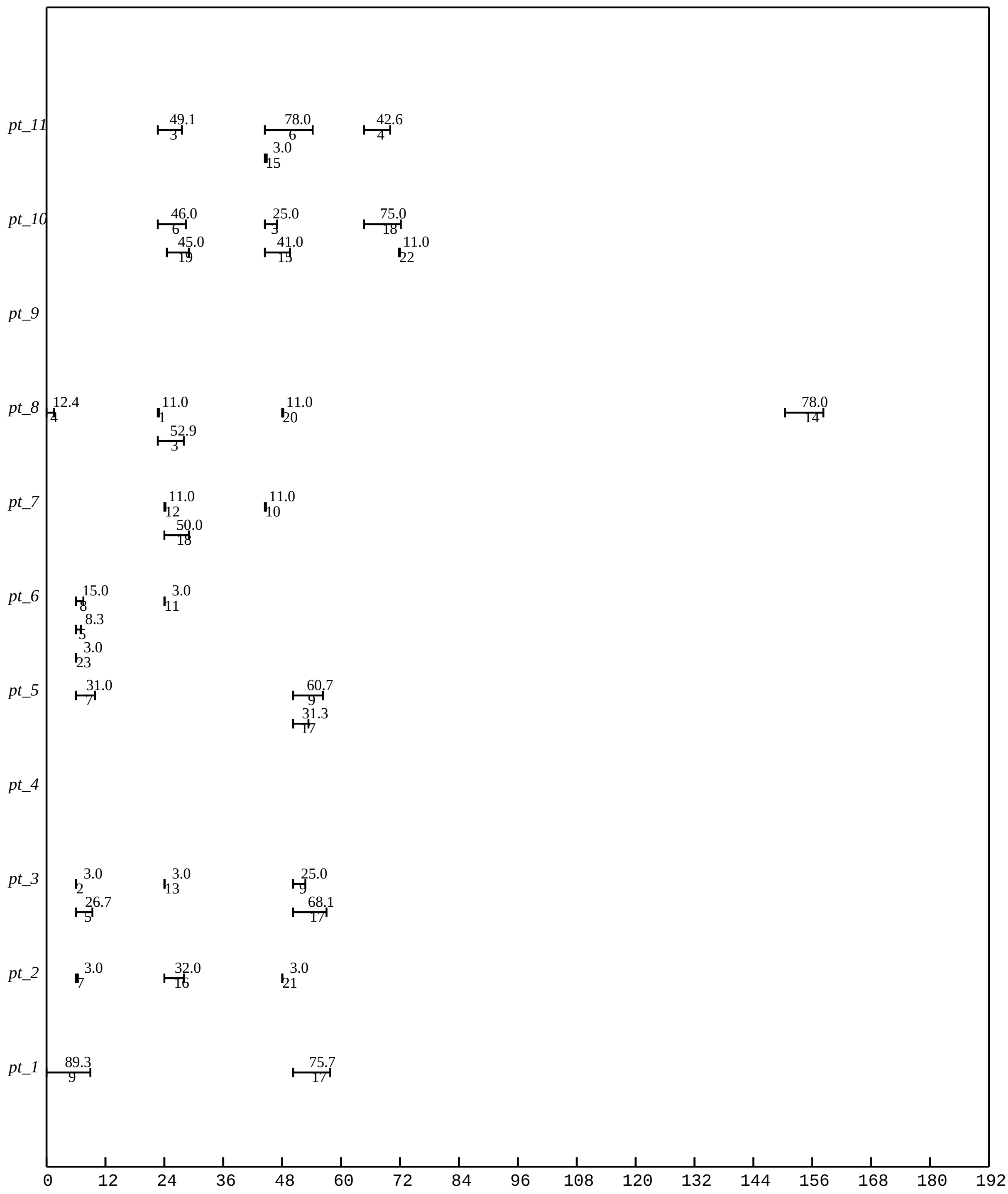


Figure 4.3: Gantt Chart for the Example with 23 Orders

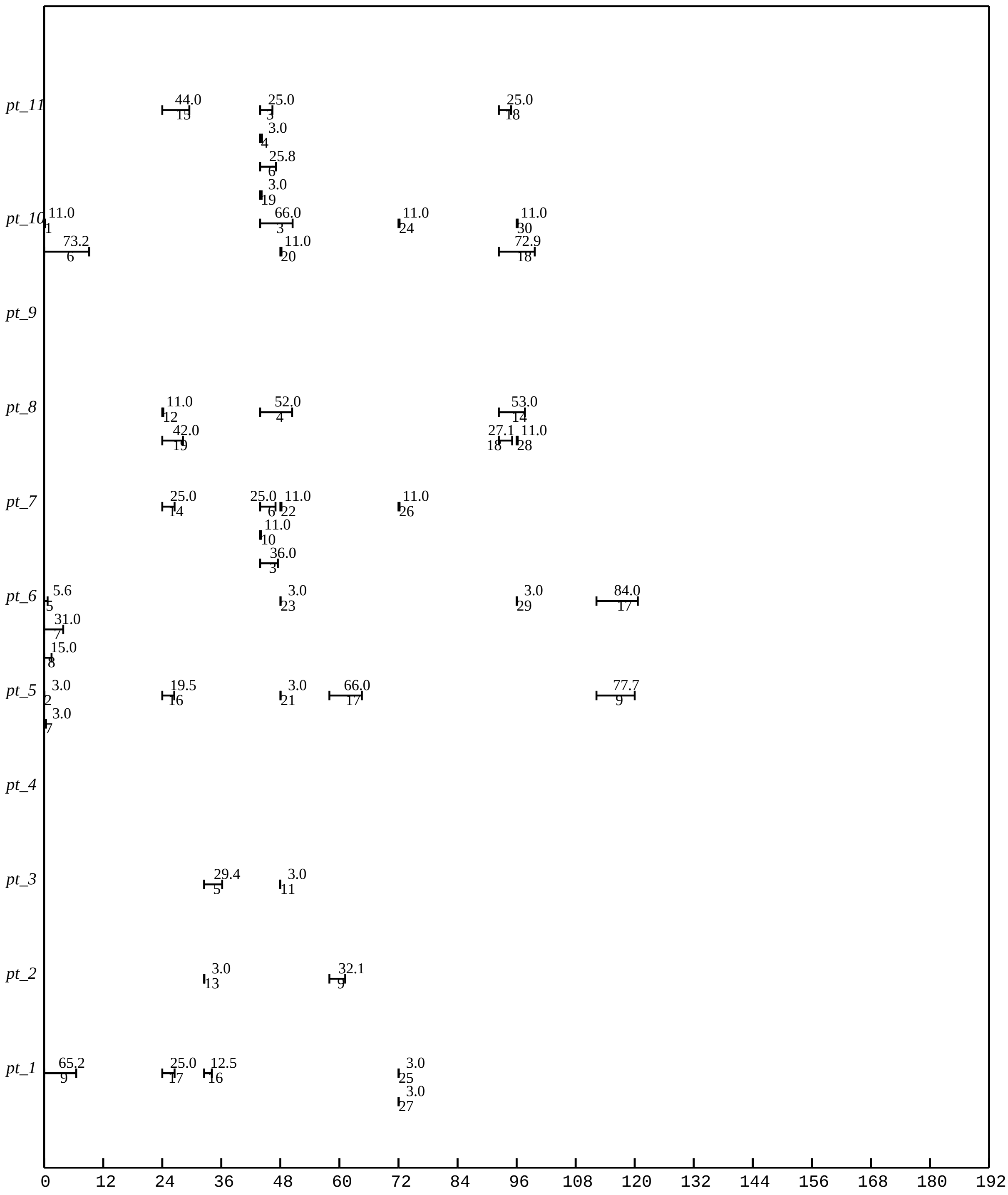


Figure 4.4: Gantt Chart for the Example with 30 Orders

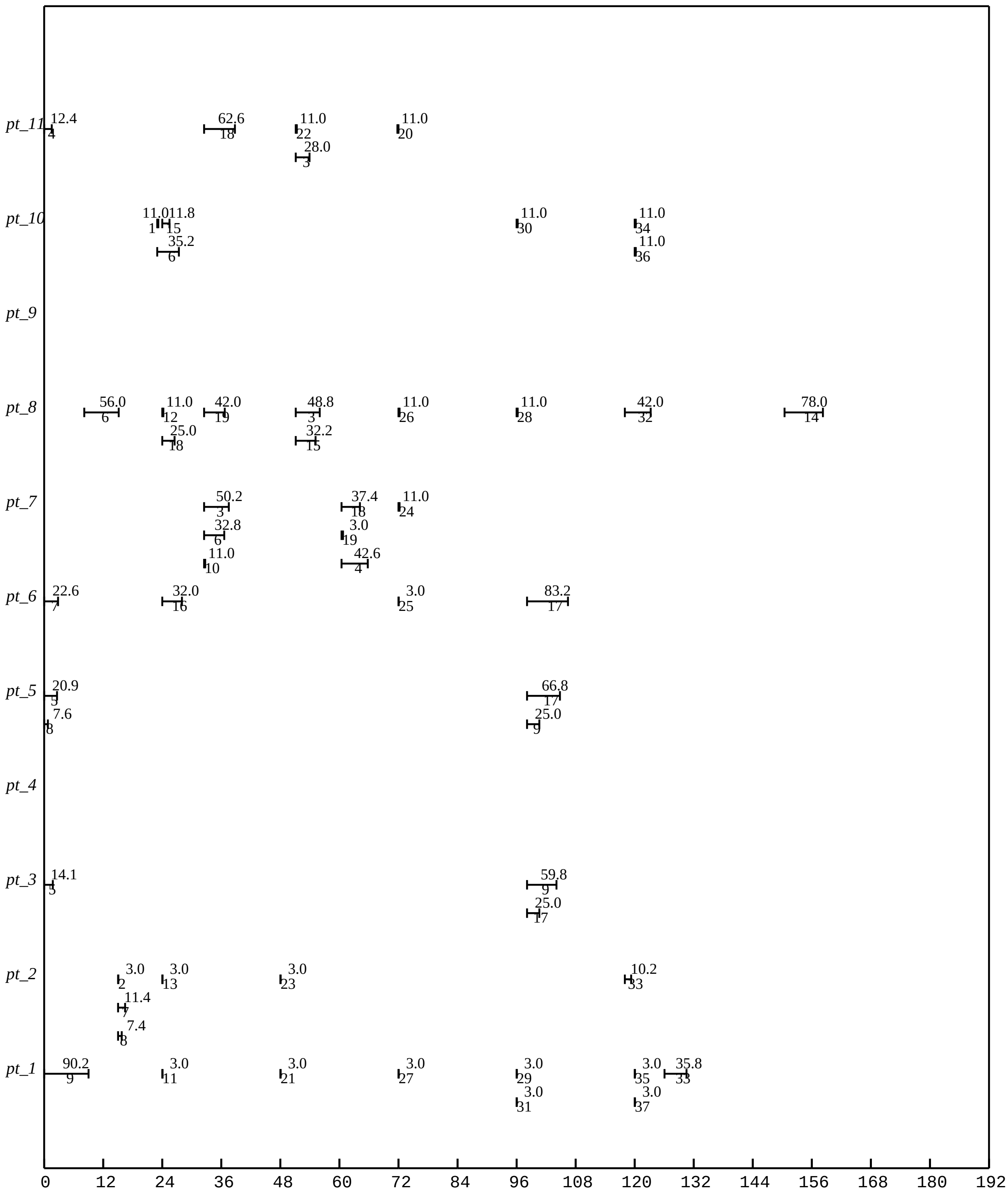


Figure 4.5: Gantt Chart for the Example with 37 Orders

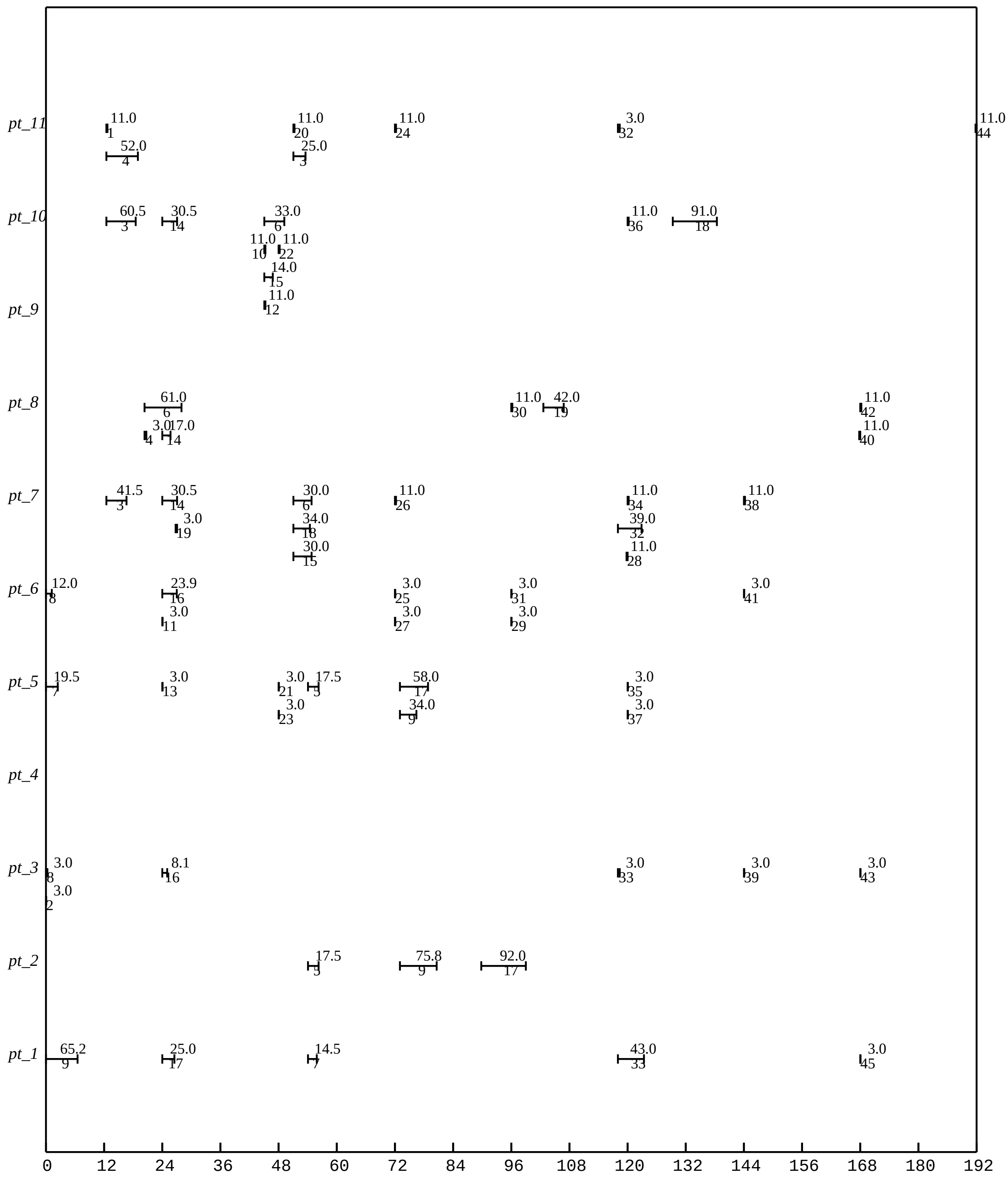


Figure 4.6: Gantt Chart for the Example with 45 Orders

Chapter 5

Multiobjective Robust Optimization

5.1 Introduction

In the literature, there has been a significant amount of work devoted in the area of short-term scheduling, which involves the determination of the order in which tasks use units and various resources and the detailed timing of the execution of all tasks so as to achieve desired performance. Most of the proposed approaches can be classified into two broad categories based on the time representation they adopt. Early attempts were mainly concentrated on the discrete time or time-indexed formulation, where the time horizon is divided into a number of intervals of equal duration. Following this approach leads to large-scale mixed integer linear programming (MILPs) involving large number of binary variables[63].

In recent years, a number of publications have focused on developing efficient methods based on continuous time representation. Two alternative continuous time approaches have been developed: the first representation requires that each task can only take place between two time points as proposed by Schilling and Pantelides[100], Mockus and Reklaitis[72], while the other representation only requires each task to start at a time point as proposed by Castro et al[21]. An event-point based approach is proposed by Ierapetritou and Floudas[51,52] and applied to the scheduling of multipurpose batch and continuous processes. Most recently, Maravelias and Grossmann[70] presented a novel continuous time MILP model for general multipurpose batch plants scheduling, which accounts for batch mixing/splitting, resource constraints, variable batch sizes and processing times, and various storage policies.

In all these studies, the problem data are assumed to be deterministic. However, in real plants, parameters like raw material availability, processing times, and market requirements vary with respect to time and are often subject to unexpected deviations. Therefore, the consideration of uncertainty in scheduling problem becomes of great importance in order to preserve plant feasibility and viability during operations.

Although there has been a substantial amount of work addressing the problem of design and planning under uncertainty, a detailed literature review of which can be found in Cheng et al.[22], the issue of uncertainty in scheduling problems has received relatively little attention. Existing work mainly includes stochastic programming approaches involving chance constraints and two-stage programming[18,58], as well as robust optimization methods[13,68,111]. A brief overview of these approaches are presented here. Lin et al.[68] proposed a robust optimization method to address the problem of scheduling with uncertain processing times, market demands, or prices. The robust optimization model was derived from its deterministic model considering the worst-case values of the uncertain parameters, and a certain infeasibility tolerance was introduced to allow constraint violations. Vin and Ierapetritou[111] addressed the problem of multiproduct batch plant scheduling under demand uncertainty. They introduced a robustness metric based on deviations from the expected performance including the infeasible scenarios. Robust schedules are generated based on a multiperiod approach. Balasubramanian and Grossmann[6] considered uncertain processing times in scheduling of multistage flowshop plants. They also proposed a multiperiod MILP model and proposed a special branch and bound algorithm with aggregated probability model to select the sequence of jobs with minimum expected makespan. Recently, Bonfill et al.[18] used a two-stage stochastic approach to address the robustness in scheduling batch processes with uncertain operation times. The objective is to minimize a weighted combination of the expected makespan and wait times. Basset et al.[13] proposed a framework considering uncertainties in processing times, equipment reliability, process yields, demands and manpower changes. They generate random instances by Monte Carlo sampling, and determine the schedules for these instances. The solutions are then analyzed to derive a number of operating policies. Orcun et al. [78] presented an approach to deal with uncertain processing times in batch processes and utilized chance constraints to take into consideration the violation of operation times constraints under certain conditions. In our earlier work[58], we developed a branch-and-bound solution framework to determine a set of alternative schedules for a given range of uncertain parameters. The idea of inference based sensitivity analysis for MILP problems was employed that has the advantage of not substantially increasing the complexity compared with the deterministic formulation.

In this chapter, we propose a multiobjective robust optimization approach. Mulvey et al.[75] developed the concept of robust optimization (RO) to handle the trade-off associated with solution and model robustness. A solution to an optimization is considered to be solution robust if it remains close to the optimal for all scenarios, and model robust if it remains feasible for most scenarios.

A wide variety of problems arising in design and operation of engineering systems require simultaneous optimization of more than one objective function. A solution that optimizes all the objectives most likely doesn't exist, thus we need to find out solutions that trade-off the different objectives. This type of problems are known as multiobjective, multicriteria or vector optimization problems, which consist of two or more conflicting objective functions with a set of constraints taken into consideration. Optimization of these problems is to identify the set of Pareto optimal solutions. A solution is Pareto optimal if improvement in one objective can only be achieved at the expense of some other objectives. In mathematical terms, for a general multiobjective optimization problem:

$$(MOP) \qquad \min_{x \in C} \quad F(x) = \begin{pmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{pmatrix} \quad n \geq 2$$

where

$$C = \{x : h(x) = 0, g(x) \leq 0, a \leq x \leq b\}$$

A point $x^* \in C$ is Pareto optimal (or non-dominated) for multiobjective optimization problem (MOP) if and only if there is no $x \in C$ such that $f_i(x) \leq f_i(x^*)$ for all $i \in 1, 2, ..., n$, with at least one strict inequality.

Classical approaches for MOP are the weighting method[113] and the ϵ-constraint method[46]. Weighting method minimizes a positively weighted sum of the individual objectives, where the choice of appropriate weighting coefficients is left to the users. For this method the objective takes the following form:

$$\sum_i \alpha_i f_i(x), \qquad \alpha_i > 0, i = 1, 2, ..., n$$

where α_i are the weights for the different objectives. ϵ-constraint method[46] minimizes a primary objective $f_p(x)$, and constrains the upper bounds for the remaining objectives as follows:

$$\begin{array}{llll} & & \min_{x \in C} & f_p(x) \\ \text{subject to} & f_i(x) \leq \epsilon_i & i = 1, ..., n & i \neq p \end{array}$$

Hillermeier[48] proposed a homotopy method that considers the set of Pareto candidates as a differentiable manifold and constructs a local chart which is fitted to the local geometry of that Pareto manifold. New Pareto candidates are generated by evaluating the local chart numerically. The normal boundary intersection (NBI)[26] method uses a geometrically intuitive parametrization to produce an even distributed set of points on the Pareto surface, even for poorly scaled problems. This method is utilized in this chapter to generate the Pareto surface of multiobjective scheduling problem. The details of this approach are provided in section 5.3 after the presentation of the proposed robust optimization model for short term scheduling in the next section. Section 5.4 is used to present the findings and discuss the characteristics of scheduling problems through the solution of three case studies whereas section 5.5 summarizes the work and present some of the ideas for future developments.

5.2 Multiobjective Robust Optimization Model

5.2.1 Deterministic Scheduling Formulation

In this chapter, the mathematical model for batch plant scheduling proposed by Ierapetritou and Floudas[51] is adopted. It follows the main idea of event based continuous time formulation and involves the following constraints:

$$\text{minimize} \quad H \quad \text{or} \quad \text{maximize} \quad \sum_s \sum_n price(s) d(s,n) \tag{5.1}$$

$$\text{subject} \quad \text{to} \quad \sum_{i \in I_j} wv(i,j,n) \leq 1 \tag{5.2}$$

$$st(s,n) = st(s,n-1) - d(s,n)$$

$$+ \sum_{i \in I_j} \rho^p(s,i) \sum_{j \in J_i} b(i,j,n-1) + \sum_{i \in I_j} \rho^c(s,i) \sum_{j \in J_i} b(i,j,n) \tag{5.3}$$

$$st(s,n) \leq stmax(s) \tag{5.4}$$

$$Vmin(i,j) wv(i,j,n) \leq b(i,j,n) \leq Vmax(i,j) wv(i,j,n) \tag{5.5}$$

$$\sum_n d(s,n) \geq r(s) \tag{5.6}$$

$$Tf(i,j,n) = Ts(i,j,n) + \alpha(i,j) wv(i,j,n) + \beta(i,j) b(i,j,n) \tag{5.7}$$

$$Ts(i,j,n+1) \geq Tf(i,j,n) - U(1 - wv(i,j,n)) \tag{5.8}$$

$$Ts(i,j,n) \geq Tf(i',j,n) - U(1 - wv(i',j,n)) \tag{5.9}$$

$$Ts(i,j,n) \geq Tf(i',j',n) - U(1 - wv(i',j',n)) \tag{5.10}$$

$$Ts(i,j,n+1) \geq Ts(i,j,n) \tag{5.11}$$

$$Tf(i,j,n+1) \geq Tf(i,j,n) \tag{5.12}$$

$$Tf(i,j,n) \leq H \tag{5.13}$$

$$Ts(i,j,n) \leq H \tag{5.14}$$

In the above formulation, allocation constraints (5.2) state that only one of the tasks can be performed in each unit at an event point (n). Constraints (5.3) represent the material balances for each state (s) expressing that at each event point (n) the amount $st(s,n)$ is equal to that at event point ($n-1$), adjusted by any amounts produced and consumed between event points ($n-1$) and (n), and delivered to the market at event point (n). The storage and capacity limitations of production units are expressed by constraints (5.4) and (5.5). Constraints (5.6) are written to satisfy the demands of final products. Constraints (5.7) to (5.14) represent time limitations due to task duration and sequence requirements in the same or different production units. Parameters $\alpha(i,j)$ and $\beta(i,j)$ are defined as: $\alpha(i,j) = \frac{2}{3}\overline{T}(i,j)$, $\beta(i,j) = \frac{2}{3}\overline{T}(i,j)/(Vmax(i,j) - Vmin(i,j))$, where $\overline{T}(i,j)$ is mean processing time of task (i) in unit (j). This is based on the assumption that there is 33% variability of the processing time around the mean value to accommodate different batch sizes, although different processing times functions can be easily adapted. When $wv(i,j,n)$ equals to 0, the last two terms in constraints (5.7) are equal to zero due to capacity constraints. Otherwise, the last two terms are added to $Ts(i,j,n)$. Therefore, the duration of task (i) at unit (j) at event point (n) depends on the amount of material being processed. The remaining timing constraints (5.8) - (5.14) represent the production recipe constraints and should be satisfied to impose the correct task sequence.

There is a lot of discussion in the literature recently regarding different modeling attempts of the deterministic scheduling problem. Maravelias and Grossmann[70] discussed different time representation schemes and proposed a general continuous time MILP formulation for the short-term scheduling of multipurposes batch plants. In this chapter, we select the above presented model since it has been shown to perform well for different case studies. However, as will become apparent in the following section, the approach presented in this chapter to address the issue of uncertainty can be utilized independent of the scheduling formulation adopted.

5.2.2 Robust Optimization

In the proposed model, demand uncertainty is described by a number of scenarios (k), each of which is associated with probability p^k. The optimal schedule of the deterministic scheduling formulation presented in the previous subsection will be robust with respect to optimality if it remains close to the optimal solution for any realization of scenario $k \in K$. This solution is called *solution robust.* The schedule is also robust with respect to feasibility if it remains almost feasible for any realization of k, which is called *model robust.* Our aim is to find robust schedules in the face of uncertainty that can help the decision maker to select the optimal solution.

In order to incorporate these two objectives, a multiobjective robust optimization formulation is proposed, which has the following form for the case of uncertain demands:

$$\text{minimize} \quad \begin{pmatrix} \sum_k p^k H^k \\ \sum_k p^k \sum_s slack^k(s) \\ \sum_k p^k \delta^k \end{pmatrix} \tag{5.15}$$

$$\text{subject to} \quad \sum_{i \in I_j} wv(i,j,n) \leq 1 \tag{5.16}$$

$$st^k(s,n) = st^k(s,n-1) - d^k(s,n)$$

$$+ \sum_{i \in I_j} \rho^p(s,i) \sum_{j \in J_i} b^k(i,j,n-1) + \sum_{i \in I_j} \rho^c(s,i) \sum_{j \in J_i} b^k(i,j,n) \tag{5.17}$$

$$st^k(s,n) \leq st^{max}(s) \tag{5.18}$$

$$Vmin(i,j)wv(i,j,n) \leq b^k(i,j,n) \leq Vmax(i,j)wv(i,j,n) \tag{5.19}$$

$$\sum_n d^k(s,n) + slack^k(s) \geq r^k(s) \tag{5.20}$$

$$Tf^k(i,j,n) = Ts^k(i,j,n) + \alpha(i,j)wv(i,j,n) + \beta(i,j)b^k(i,j,n) \tag{5.21}$$

$$Ts^k(i,j,n+1) \geq Tf^k(i,j,n) - U(1 - wv(i,j,n)) \tag{5.22}$$

$$Ts^k(i,j,n) \geq Tf^k(i^{'},j,n) - U(1 - wv(i^{'},j,n)) \tag{5.23}$$

$$Ts^k(i,j,n) \geq Tf^k(i^{'},j^{'},n) - U(1 - wv(i^{'},j^{'},n)) \tag{5.24}$$

$$Ts^k(i,j,n+1) \geq Ts^k(i,j,n) \tag{5.25}$$

$$Tf^k(i,j,n+1) \geq Tf^k(i,j,n) \tag{5.26}$$

$$Tf^k(i,j,n) \leq H^k \tag{5.27}$$

$$\delta^k \geq H^k - \sum_k p^k H^k, \quad \delta^k \geq 0, \quad H^k \leq U \tag{5.28}$$

The first objective is minimizing the expected makespan, which is derived from the original objective in deterministic formulation. Model robustness is represented by the second objective that minimizes the expected unsatisfied demands, which is computed by introducing the artificial variables ($slack^k(s)$) in the demand constraints (5.20). The concept of upper partial mean (UPM) introduced by Ahmed and Sahinidis[2] is used in the third objective function in order to optimize the solution robustness. They define the variance measure $\overline{\delta}$ as follows:

$$\overline{\delta} = \sum_k p^k \delta^k, \quad \delta^k = max(0, H^k - \sum_k p^k H^k)$$

where δ^k corresponds to the positive deviation of makespan under scenario (k) from the expected value. The main advantage of using UPM instead of variance is that it can avoid introducing nonlinearities in the formulation. Thus, the resulting model remains a mixed-integer linear programming (MILP) problem.

Comparing to the deterministic problem presented in section 5.2.1, in this formulation, the binary variables $wv(i, j, n)$ that represent the task sequences remain the same over all scenarios, while the continuous variables that correspond to the batch sizes, and the starting and finishing times can vary to accommodate the realization of different scenarios. Thus, the schedules obtained by solving this multiobjective optimization problem include robust assignments that can accommodate the demand uncertainty.

Note that this robust optimization model is written for a general batch plant scheduling problem where the objective is to minimize the makespan. However, other scheduling problems can have different objectives and constraints, as will be shown by the last case study in section 4 for the problem of crude oil unloading. In these cases, the above formulation has to be modified to accommodate the different objectives.

5.3 Multiobjective Optimization

5.3.1 Normal Boundary Intersection

NBI is a solution methodology developed by Das and Dennis[26] for generating Pareto surface in nonlinear multiobjective optimization problems. It is proved that this method is independent of the relative scales of the objective functions and is successful in producing an evenly distributed set of points in the Pareto surface given an evenly distributed set of parameters, which is an advantage compared to the most common multiobjective approaches - weighting method and the ϵ-constraint method.

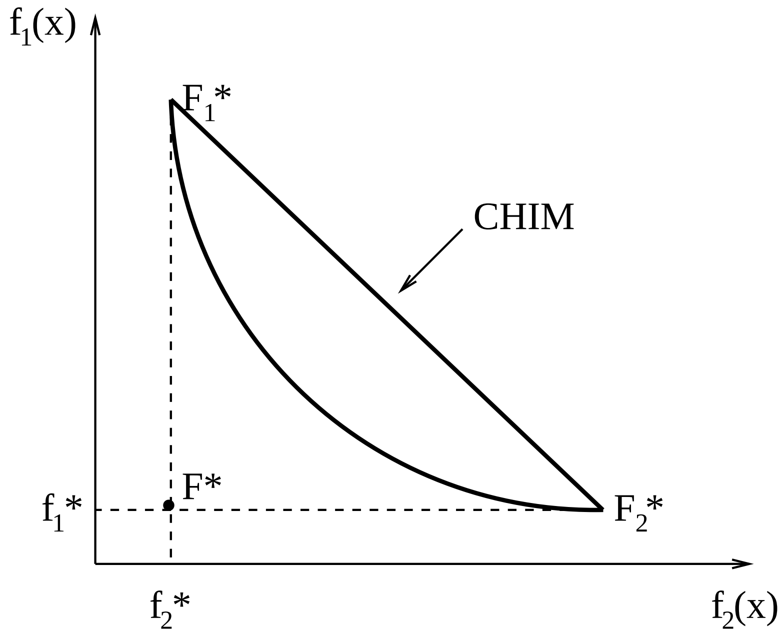


Figure 5.1: Pareto Optimal Surface for a Two-objective Problem

The *anchor point* for the (MOP) described in section 1, F_i^*, is obtained when the *ith* objective is minimized independently, while f_i^* represents the individual minima of the *ith* objective. A two-objective case is illustrated in Figure 5.1. The shadow minimum (*utopia point*) F^*, is defined as the vector containing the individual global minima of the objectives, i.e.

$$F^* = \begin{pmatrix} f_1^* \\ f_2^* \\ \vdots \\ f_n^* \end{pmatrix}$$

The convex hull of individual minima (CHIM) has the following definition: let x_i^* be the respective minimizer of $f_i(x), i = 1, ...n$ for $x \in C$. Let $F_i^* = F(x_i^*), i = 1, ..., n$, Φ be the $n \times n$ matrix whose i^{th} column is $F_i^* - F^*$. Then the set of points in R^n that are convex combinations of $F_i^* - F^*$, i.e., $\Phi\omega : \omega \in R^n, \sum_i \omega_i = 1, \omega_i \geq 0$ is referred to as the CHIM. The set of attainable objective vectors: $F(x) : x \in C$ is denoted by

$\mathcal{F}$ so C is mapped onto $\mathcal{F}$ by F. The space R^n which contains $\mathcal{F}$ is referred to as objective space. The boundary of $\mathcal{F}$ is denoted by $\partial\mathcal{F}$.

NBI method determines the portion of $\partial\mathcal{F}$ which contains the Pareto optimal points. The principal idea behind this approach is that the intersection point between the boundary $\partial\mathcal{F}$ and the normal pointing towards the origin emanating from any point in the CHIM is a point on the portion of $\partial\mathcal{F}$ containing the efficient points. This point is guaranteed to be a Pareto optimal point if the trade-off surface in the objective space is convex.

Given a convex weighting ω, $\Phi\omega$ represents a point in the CHIM. Let $\hat{n}$ denote the unit normal to the CHIM simplex towards the origin; then $\Phi\omega + t\hat{n}$ represents the set of points on that normal. The point of intersection of the normal and the boundary of $\mathcal{F}$ closest to the origin is the global solution of the following subproblem:

$$
\begin{aligned}
(NBI_\omega) \qquad & \max_{x,t} \; t \\
\text{subject to} \qquad & \Phi\omega + \hat{t}n = F(x) \\
& h(x) = 0 \\
& g(x) \leq 0 \\
& a \leq x \leq b
\end{aligned}
$$

The vector constraint $\Phi\omega + \hat{t}n = F(x)$ ensures that the point x is actually mapped by F to a point on the normal, while the remaining constraints ensure feasibility of x with respect to the original problem (MOP). Note that if the shadow minimum F^* is not the origin, then the first set of constraint should be $\Phi\omega + \hat{t}n = F(x) - F^*$. The i^{th} column of Φ is defined as: $\Phi(:,i) = F(x_i^*) - F^*$. Since $f_i(x_i^*) = f_i^*$, clearly, $\Phi(i,i) = 0$. Moreover, since x_i^* is the minimizer of $f_i(x)$ over $x_j, j = 1, ..., n$, $\Phi(j,i) \geq 0, j \neq i$. By solving NBI_ω for various ω, a number of points on the boundary of $\mathcal{F}$ are obtained thus effectively constructing the Pareto surface.

Quasi-normal direction is used instead of normal direction so that it represents equally weighted linear combination of columns of Φ, multiplied by -1 to ensure that it points towards the origin. Specifically,

$$\hat{n} = -\Phi v$$

where v is a fixed vector with strictly positive components. The quasi-normal $\hat{n}$ defined above has the property that the NBI point found for a certain ω is completely independent of the scales of the objective functions. Since Φ has non-negative components, $\hat{n}$ has non-positive components. When the objective functions $f_i(x)$ are scaled by arbitrary factors s_i,

$$f_i(x) \Longrightarrow s_i f_i(x), i = 1, ..., n.$$

it can be proved that the unscaled problem and its scaled version correspond to the same solution[26]. Here $\hat{n}$ is chosen to be $-\Phi e$, where e is the column vector of all ones.

In order to generate different values of ω, let's assume that for a n-objective problem, δ_j is the uniform spacing between two consecutive ω_j values for $j = 1, ...n-1$. The possible values that can be taken by ω_1 are: $[0, \delta_1, 2\delta_1, ..., 1]$. Given a particular value of ω_1, the possible values of ω_2 corresponding to that value of ω_1 are: $[0, \delta_2, 2\delta_2, ..., k_2\delta_2]$, where $k_2 = I[\frac{1-\omega_1}{\delta_2}]$. Hence, corresponding to $\omega_i, i = 1, ..., j-1$, the possible values of ω_j for $j = 1, ..., n-1$ are: $[0, \delta_j, 2\delta_j, ..., k_j\delta_j]$, where $k_j = I[\frac{1-\sum_i \omega_i}{\delta_j}]$. Finally, the last component of ω_n is defined as: $\omega_n = 1 - \sum_{i=1} \omega_i$.

The basic steps of NBI in the context of scheduling problems are presented in the next section.

5.3.2 Robust Scheduling

Step 1: Determine the anchor points.

The robust optimization model for scheduling problems as presented in section 5.2.2 has three objectives, which are the expected value of makespan, unsatisfied demand (model robustness), and the upper partial mean of the makespan (solution robustness). In order to determine the anchor points, the robust optimization formulation is solved with one objective function being minimized each time. The expected makespan, model robustness, and solution robustness is minimized with respect to constraints (5.16) - (5.28) individually, and the minimum value and the values of the other two objectives are saved. Since the makespan requirement is imposed through the inequality in constraint (5.27), when the problem is solved to minimize the model robustness or solution robustness, the makespan that corresponds to each scenario H^k obtained may not be equal to the finishing time of the last task. Thus, in order to get the optimal value of expected makespan at the anchor points, if model or solution robustness is optimized first, the following step is required.

Step 2: Tighten the anchor points.

When model or solution robustness is minimized first, they are fixed at the optimal values and the problem of minimizing the expected makespan is solved again. Thus, the resulting points are the real anchor points that contain the optimal value of the expected makespan corresponding to the finishing time of the last performed task. Then the utopia point F^* and the matrix Φ can be correctly determined.

Step 3: Formulate and solve problem (NBI_ω) iteratively for different values of ω.

For the case of robust formulation of section 5.2.2, the problem takes the following form:

$$\begin{aligned} & \text{maximize} && t \\ \text{subject to} & \text{constraints}(5.16) - (5.28) \\ \Phi \begin{pmatrix} \omega_1 - t \\ \omega_2 - t \\ \omega_3 - t \end{pmatrix} &= \begin{pmatrix} \sum_k p^k H^k \\ \sum_k p^k \sum_s slack^k(s) \\ \sum_k p^k \delta^k \end{pmatrix} - F^* \end{aligned}$$

A number of points that are evenly distributed on the Pareto optimal surface are then obtained. The number of points depends on the selected step size to generate ω. Each of the points represents a trade-off solution between the expected performance, feasibility and deviation from the mean.

In the next section, two scheduling problems are presented to illustrate the results of the proposed approach. Both of them follow the robust model section 5.2.2.

5.4 Case Studies

Example 1: The first example involves a single product production line consisting of a mixer, a reactor and a purifier as shown in Figure 5.2. The data for this example are obtained from Ierapetritou and Floudas[51]. We added an additional requirement of satisfying at least 10% of the demand values so that the production will not be zero when the makespan is minimized. The demand is considered to be the uncertain parameter varying within the range of $[20, 100]$.

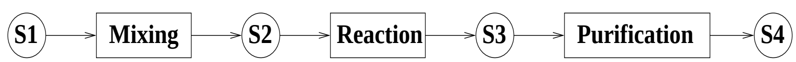


Figure 5.2: State-Task Network Representation for Example 1

The problem is formulated as a three-objective robust optimization problem considering 5 demand scenarios $(20, 40, 60, 80, 100)$. The following points are obtained by minimizing the objectives individually:

$$f_1(x^*) = \begin{pmatrix} 6.46 \\ 54 \\ 0.092 \end{pmatrix} \quad f_2(x^*) = \begin{pmatrix} 11.5 \\ 0 \\ 0.662 \end{pmatrix} \quad f_3(x^*) = \begin{pmatrix} 6.766 \\ 50 \\ 0 \end{pmatrix}$$

Therefore, the shadow minimum is $F^* = \begin{pmatrix} 6.46 \\ 0 \\ 0 \end{pmatrix}$, and the matrix $\Phi = \begin{pmatrix} 0 & 4.136 & 0.306 \\ 54 & 0 & 50 \\ 0.092 & 0.662 & 0 \end{pmatrix}$

Selecting the step sizes to be $\delta_1 = 0.1$ and $\delta_2 = 0.05$, the possible values of ω_1 are:

$$[0, 0.1, 0.2, ..., 0.9, 1]$$

Since $\frac{1-\omega_1}{\delta_2}$ must be an integer, the possible values of ω_2 are:

$$[0, 0.05, 0.1, ..., 1-\omega_1]$$

ω_3 can be easily calculated by $\omega_3 = 1 - \omega_1 - \omega_2$.

NBI_ω problem is then formulated as follows:

$$\begin{aligned} \text{maximize} \quad & t \\ \text{subject to} \quad & \text{constraints}(5.16) - (5.28) \end{aligned}$$

$$\begin{pmatrix} 0 & 4.136 & 0.306 \\ 54 & 0 & 50 \\ 0.092 & 0.662 & 0 \end{pmatrix} \begin{pmatrix} \omega_1 - t \\ \omega_2 - t \\ \omega_3 - t \end{pmatrix} = \begin{pmatrix} \sum_k p^k H^k - 6.46 \\ \sum_k p^k \sum_s slack^k(s) \\ \sum_k p^k \delta^k \end{pmatrix}$$

By solving the NBI_ω problem repeatedly for different values of ω, we obtained the Pareto optimal surface as shown in Figure 5.3, consisting of 3 schedules as presented in Table 5.1.

In this example, it is noticed that there are four dominated points. The corresponding values of $(\omega_1, \omega_2, \omega_3)$, and the objectives of these four points are listed in Table 5.2, in which the last point is a Pareto optimal solution that dominates the four points. We observed that these points correspond to the same schedule (schedule 1), for which the maximum production is 50 determined by the maximum capacity of purifier which is 50. That explains why the production at the 5 scenarios $(20, 40, 60, 80, 100)$ is $(20, 40, 50, 50, 50)$ for all these four points, which makes the expected unmet demand 18 $(18 = [(60-50) + (80-50) + (100-50)]/5)$. We also observed that for the dominated points, the makespans H^k are larger than the real last finishing times Tf^k_{last}, which is the actual makespan, and Tf^k_{last} of those points are the same for the same scenario k. Again, this arises from the inequality constraint (5.27). The four dominated points essentially represent the same solution which corresponds to the following set of objective values $f_1 = 9.22$, which is the expected makespan, $f_2 = 18$, which represents the expected unmet demand and $f_3 = 0.43$, which is the expected positive deviation from the average makespan. This is due to

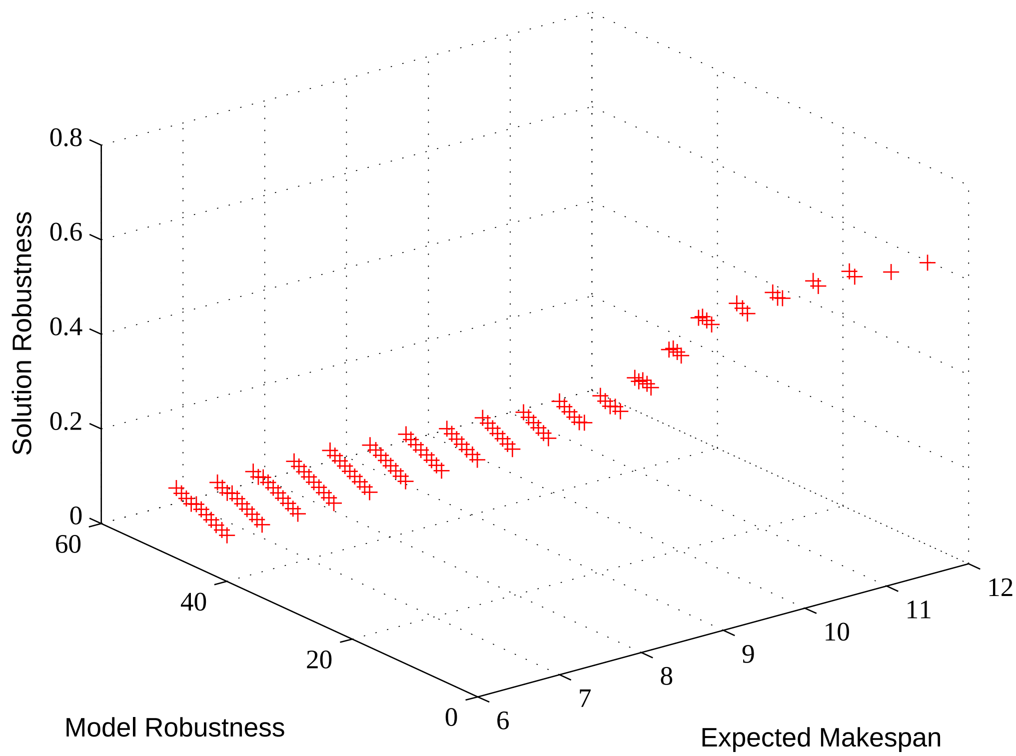


Figure 5.3: Pareto Optimal Surface for Example 1

the fact that any solution following the assignment of schedule 1 can have a maximum production of 50 due to the capacity limit of purifier, and the corresponding expected makespan is $9.22hr$, while to produce the same amount of product with the assignment of schedule 2, $10.22hr$ is required. Thus, the Pareto surface is not continuous around the region where the optimal schedule is changing from schedule 1 to schedule 2. However, the inequality in constraint (5.27) makes the formulation of NBI_ω subproblem more flexible, since it leads to a solution in the feasible region, although it might not exist a Pareto point on that direction. If constraint (5.27) is modified to enforce $H^k = Tf^k_{last}$, the NBI_ω subproblems for these four ωs in Table 5.2 will be infeasible.

To analyze more the discontinuity in the Pareto surface, we then increased the maximum capacities of all the units to 100 in order to get a *smooth* Pareto optimal

(task, unit)	n0	n1	n2
(mixing, mixer)	1	0	0
(reaction, reactor)	0	1	0
(seperation, still)	0	0	1

(Schedule 1)

(task, unit)	n0	n1	n2	n3
(mixing, mixer)	1	0	0	0
(reaction, reactor)	0	1	0	0
(seperation, still)	0	0	1	1

(Schedule 2)

(task, unit)	n0	n1	n2	n3
(mixing, mixer)	1	1	0	0
(reaction, reactor)	0	1	1	0
(seperation, still)	0	0	1	1

(Schedule 3)

Table 5.1: Values of Binary Variables of Optimal Schedules for Example 1

surface as shown in Figure 5.4. It was verified that all the points correspond to Pareto optimal solutions and one schedule (schedule 1) is found for all the points.

Example 2: The second example[51] considers two different products produced through five processing stages: heating, reactions 1, 2 and 3, and separation of product 2 from impure E as illustrated in the state-task network (STN) representation in Figure 5.5. The nominal demand for both products 1 and 2 is $(80, 80)$ and is assumed to exhibit a varibility of $\pm 50\%$. 5 scenarios $(40, 60, 80, 100, 120)$ are selected to represent the uncertain demand for each product and thus result to a total of 25 scenarios. 10% demand satisfaction is also assumed for this example.

(ω1, ω2, ω3)	(obj1, obj2, obj3)
(0, 0.65, 0.35)	(9.91, 18, 0.56)
(0.1, 0.65, 0.25)	(9.94, 18, 0.55)
(0.2, 0.65, 0.15)	(9.97, 18, 0.54)
(0.3, 0.65, 0.05)	(9.99, 18, 0.54)
(0, 0.6, 0.4)	(9.22, 18, 0.43)

Table 5.2: Dominated Points vs. Pareto Point

Following the procedure presented in section 5.3.2, the individual minimum points are obtained as follows:

$$f_1(x^*) = \begin{pmatrix} 5.055 \\ 143.65 \\ 0.045 \end{pmatrix} \quad f_2(x^*) = \begin{pmatrix} 10.56 \\ 0 \\ 0.696 \end{pmatrix} \quad f_3(x^*) = \begin{pmatrix} 5.236 \\ 143.55 \\ 0 \end{pmatrix}$$

Therefore, the shadow minimum is $F^* = \begin{pmatrix} 5.055 \\ 0 \\ 0 \end{pmatrix}$, and the matrix $\Phi = \begin{pmatrix} 0 & 5.501 & 0.181 \\ 143.65 & 0 & 143.55 \\ 0.045 & 0.696 & 0 \end{pmatrix}$

Similar to Example 1, the step sizes are chosen to be $\delta_1 = 0.1$ and $\delta_2 = 0.05$, and NBI_ω problem is formulated and solved for different values of ω. The resulting Pareto optimal surface is shown in Figure 5.6, which contains 8 different schedules. The values of the binary variables that represent the sequences of tasks being performed in each unit are listed in Table 5.3. Taking a closer look at the optimal Pareto solutions, a number of interesting observations can be made. For example, focusing on two points A and B as shown in Figure 5.6, point A which is obtained with $\omega = (0, 1, 0)$ is in the area of solutions that prefer model robustness. The three objective values for this point are $10.56, 0, 0.70$ for expected makespan, model robustness, and solution robustness, respectively, and the optimal schedule is schedule 8. On the other hand, point B represents the optimal solution for $\omega = (0.6, 0.4, 0)$, which corresponds to

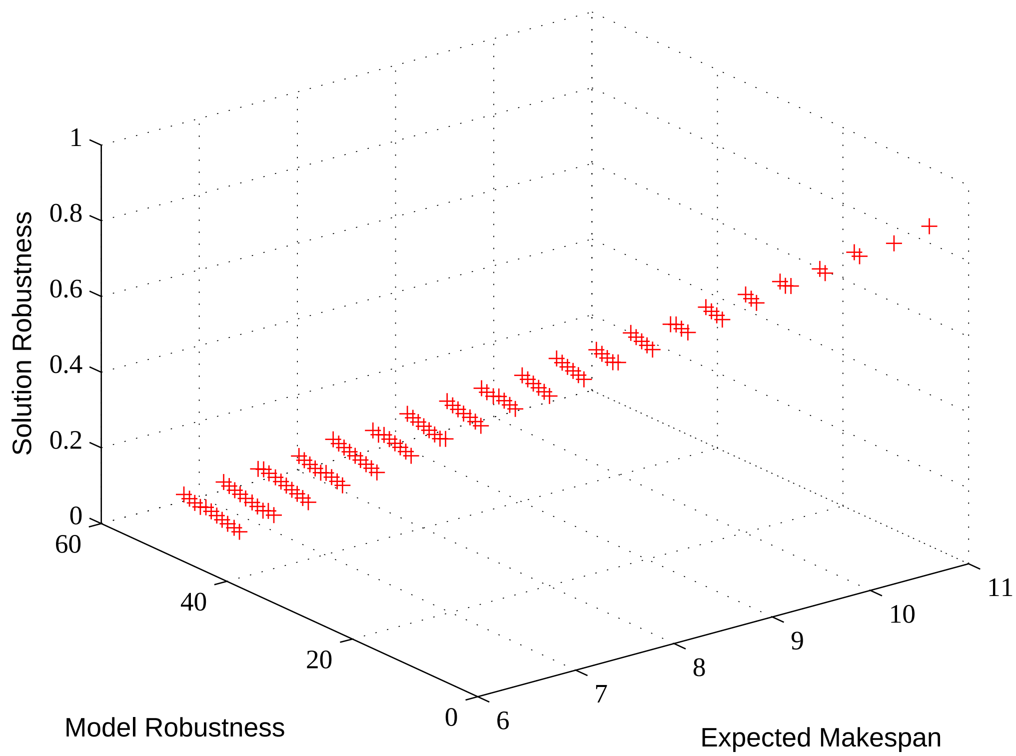


Figure 5.4: Pareto Optimal Surface for Example 1 with Increased Unit Capacities

schedule 4. Comparing to schedule 8, this solution corresponds to a decision that favors the expected makespan and solution robustness, the values of which are 6.82 and 0.25, respectively, at the expense of low model robustness, which is 64.39. For the nominal demand of $(80, 80)$, schedule 8 requires $9.83hr$ and schedule 4 prefers having a shorter makespan of $8.46hr$. However, at the maximum demand value $(120, 120)$, schedule 8 focuses more on meeting the demands and can produce the required amount within $13.32hr$, while schedule 2 results in an unsatisfied amount of 33.33 units for product 1 and 32.25 units for product 2. The difference between schedules 4 and 8 can be better illustrated with the gantt charts at the nominal demand $(80, 80)$, as shown in Figure 5.7.

The average CPU time for solving the 121 NBI subproblems is around $50min$ using GAMS/CPLEX 9.0 on a dual Pentium 4 Xeon 3.06 GHz cluster. Figure 5.8 is

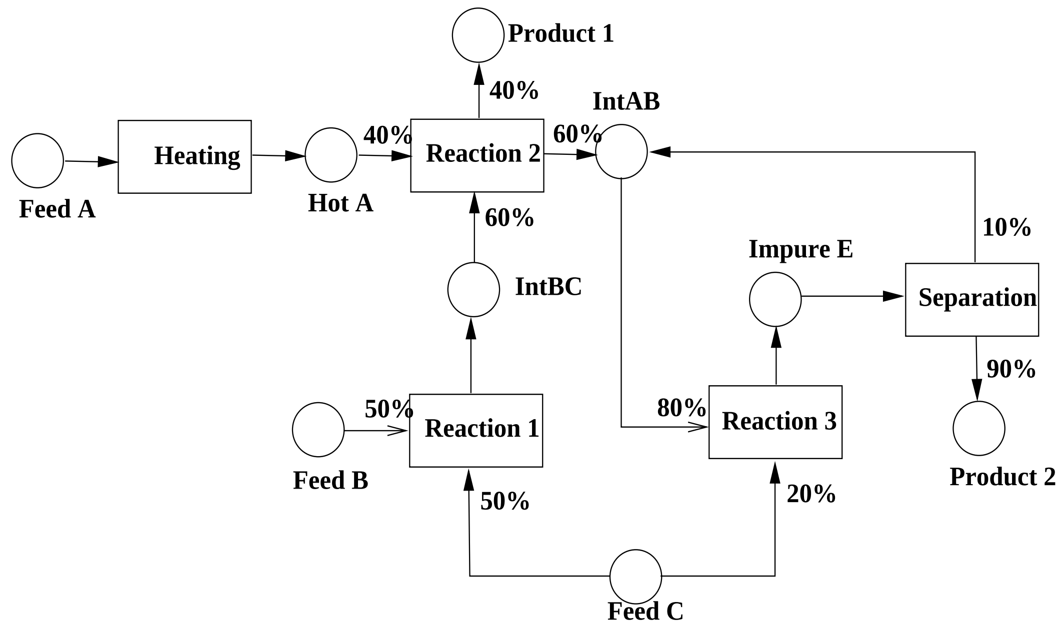


Figure 5.5: State-Task Network Representation for Example 2

a histogram showing the CPU times required to solve the subproblems.

5.5 Summary and Future Work

A multiobjective robust optimization model is proposed to deal with the problem of uncertainty in scheduling considering the expected performance, model robustness and solution robustness. NBI technique is utilized to solve the multiobjective model and successfully produce Pareto optimal surface that captures the trade-off among different objectives in the face of uncertainty. Two cases studies, which are batch plant scheduling problems of different sizes, are presented to illustrate the effectiveness of the proposed approach.

The work can be further extended to investigate the cases where preferences exist among the objectives, so as to generate more meaningful Pareto optimal solutions. When the objectives are not equally important, not all of the Pareto optimal solutions are desired. Therefore, a new method is required to consider the decision maker's preference before the entire Pareto surface is generated. This will help reducing the

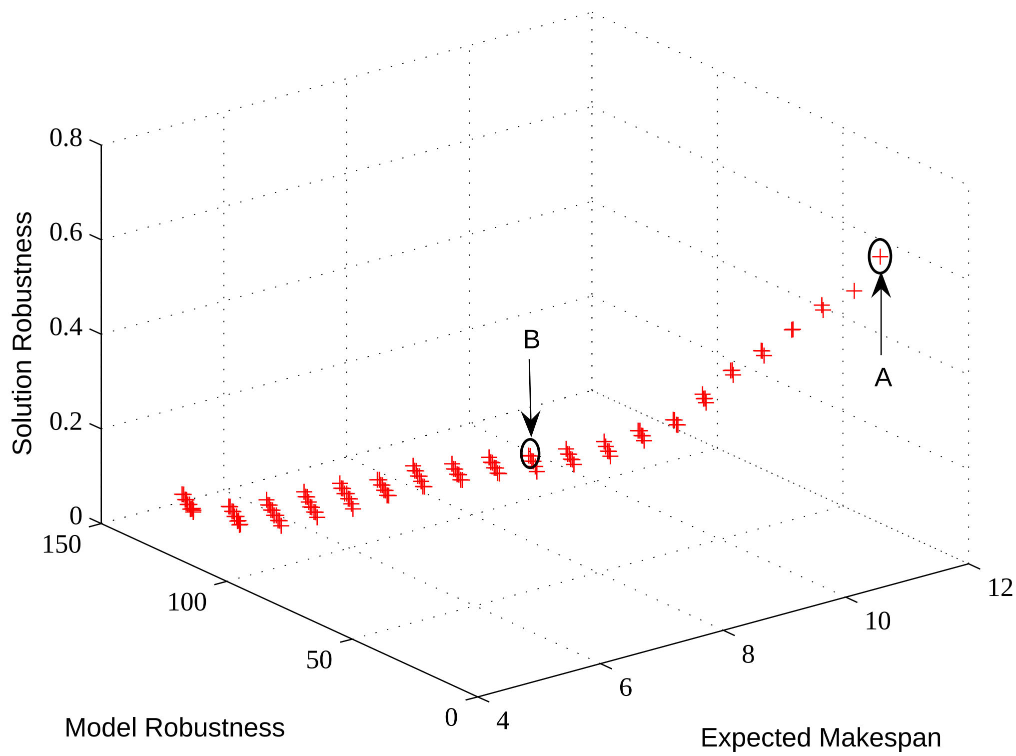


Figure 5.6: Pareto Optimal Surface for Example 2

computational complexity of the proposed approach. In addition, there are cases that instead of unique anchor points, anchor curves are found due to the fact that the objectives are not entirely conflicting with each other. For example, in the case study of crude oil unloading and mixing, the first objective is the total operation cost, which includes the sea waiting and unloading cost, inventory cost of storage tanks, and inventory cost of charging tanks. If these three costs are considered separately, when the value of one of the objectives is fixed, the anchor points of the other two objectives can form a Pareto surface. For cases where multiple anchor points exist, it will be of interest to study how the selection of anchor points can affect the Pareto surface.

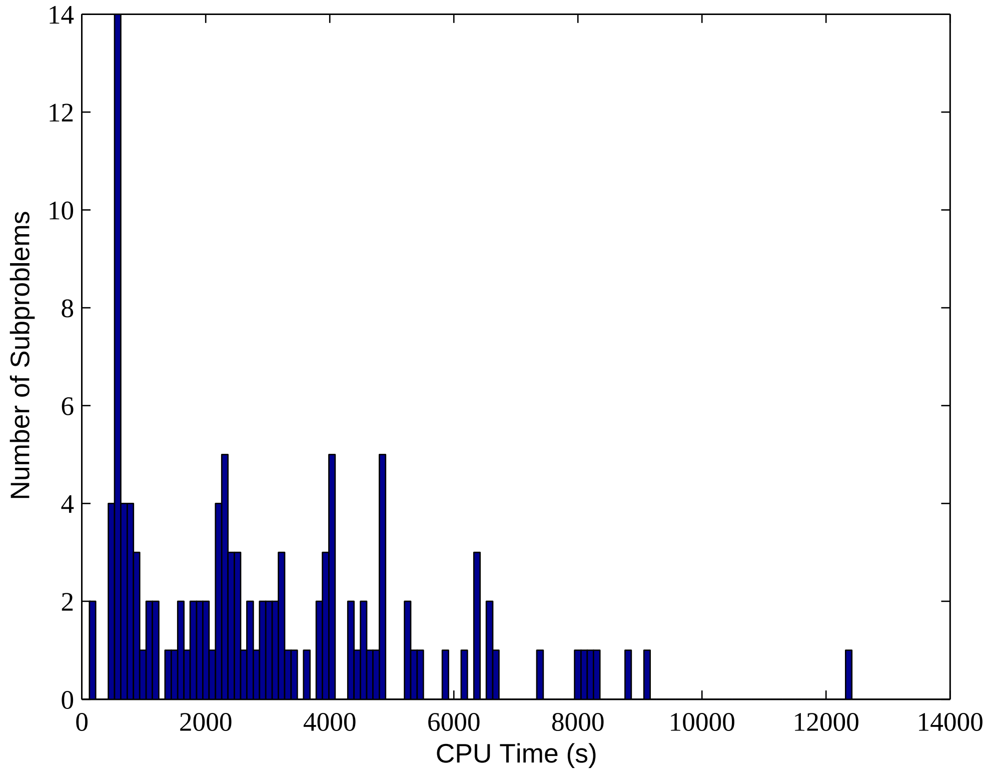


Figure 5.8: Histogram of CPU Time for solving Subproblems in Example 2

Nomenclature

Indices

i = tasks

j = units

n = event points

s = states

k = scenarios

Parameters

p^k = probability of scenario k

$\rho^p(s,i), \rho^c(s,i)$ = proportion of state s produced, consumed from task i, respectively

$r^k(s)$ = market requirement for state s at the end of the time horizon of scenario k

Vmin(i,j) = minimum capacity of unit j when processing task i

Vmax(i,j) = maximum capacity of unit j when processing task i

stmax(s) = maximum storage capacity of state s

Variables

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

$b^k(i, j, n)$ = amount of material undertaking task i in unit j at event point n of scenario k

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

H^k = time horizon of scenario k

$st^k(s, n)$ = amount of state s at event point n of scenario k

$Ts^k(i, j, n)$ = starting time of task i in unit j at event point n of scenario k

$Tf^k(i, j, n)$ = finishing time of task i in unit j at event point n scenario k

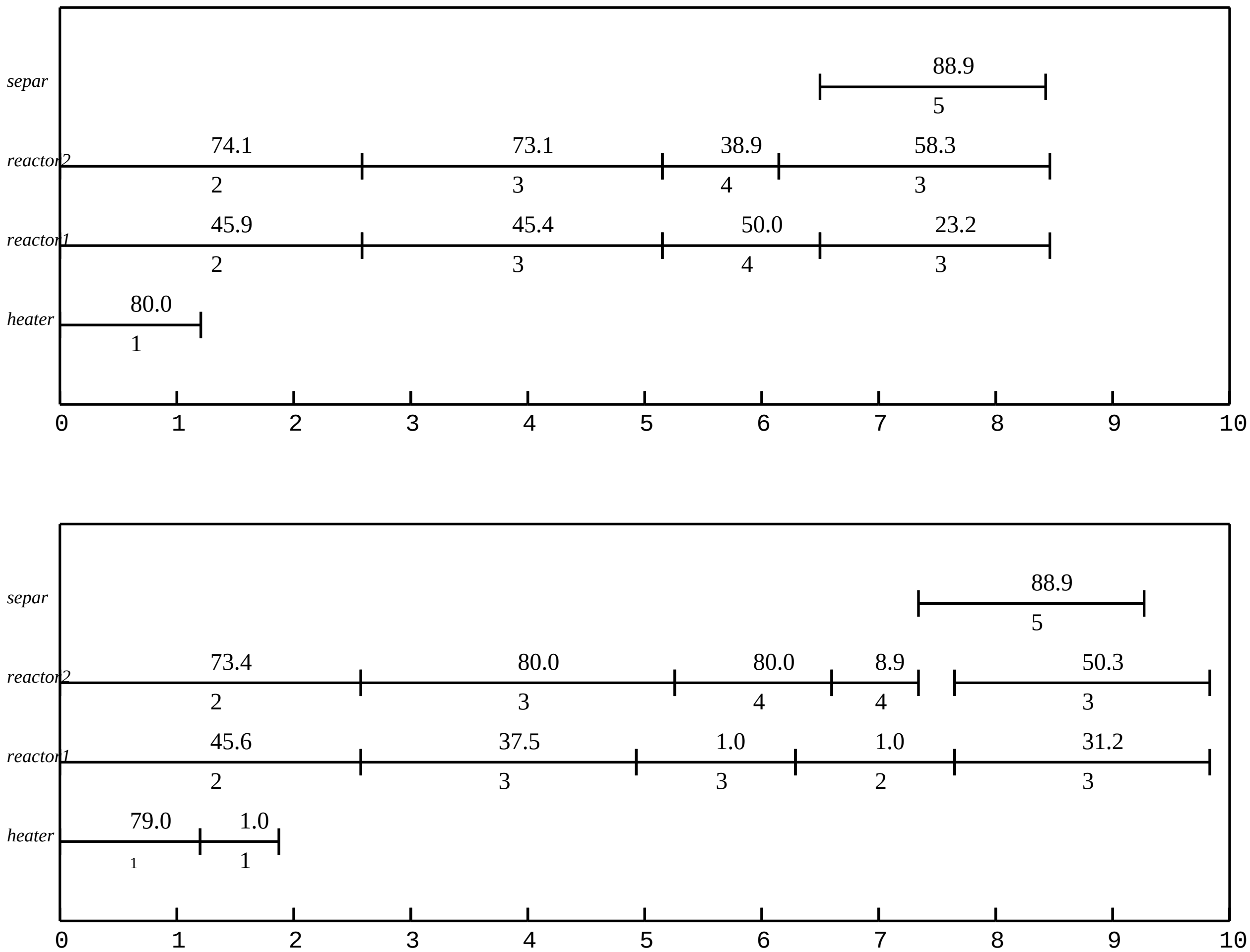


Figure 5.7: Gantt Charts of Schedules 4 (top) and 8 (bottom) at Nominal Demand

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1				
(reaction1, reactor1)	1				
(reaction1, reactor2)	1				
(reaction2, reactor1)			1	1	
(reaction2, reactor2)		1			
(reaction3, reactor1)					
(reaction3, reactor2)				1	
(seperation, still)					1

(Schedule 1)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1				
(reaction1, reactor1)				1	
(reaction1, reactor2)	1				
(reaction2, reactor1)					1
(reaction2, reactor2)		1			1
(reaction3, reactor1)					
(reaction3, reactor2)			1		
(seperation, still)				1	

(Schedule 2)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1				
(reaction1, reactor1)		1			
(reaction1, reactor2)	1				
(reaction2, reactor1)				1	
(reaction2, reactor2)		1		1	
(reaction3, reactor1)			1		
(reaction3, reactor2)					
(seperation, still)					1

(Schedule 3)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1				
(reaction1, reactor1)		1			
(reaction1, reactor2)		1			
(reaction2, reactor1)			1		1
(reaction2, reactor2)			1		1
(reaction3, reactor1)				1	
(reaction3, reactor2)				1	
(seperation, still)					1

(Schedule 4)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)		1	1		
(reaction1, reactor1)	1			1	
(reaction1, reactor2)		1			
(reaction2, reactor1)					1
(reaction2, reactor2)			1		1
(reaction3, reactor1)					
(reaction3, reactor2)				1	
(seperation, still)					1

(Schedule 5)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1	1			
(reaction1, reactor1)		1		1	
(reaction1, reactor2)	1				
(reaction2, reactor1)			1		1
(reaction2, reactor2)		1			1
(reaction3, reactor1)					
(reaction3, reactor2)			1	1	
(seperation, still)					1

(Schedule 6)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1	1			
(reaction1, reactor1)	1			1	
(reaction1, reactor2)	1				
(reaction2, reactor1)		1			1
(reaction2, reactor2)		1	1		1
(reaction3, reactor1)			1		
(reaction3, reactor2)				1	
(seperation, still)					1

(Schedule 7)

(task, unit)	n0	n1	n2	n3	n4
(heating, heater)	1	1			
(reaction1, reactor1)	1			1	
(reaction1, reactor2)	1				
(reaction2, reactor1)		1	1		1
(reaction2, reactor2)		1			1
(reaction3, reactor1)					
(reaction3, reactor2)			1	1	
(seperation, still)					1

(Schedule 8)

Table 5.3: Values of Binary Variables of Optimal Schedules for Example 2

Chapter 6

Short-Term Scheduling under Uncertainty using MILP Sensitivity Analysis

6.1 Introduction

Although there is a large number of papers to address uncertainty in process design, much less attention is devoted to the issue of uncertainty in process planning and scheduling mainly due to increased complexity of the deterministic problem. Among the work appeared in the literature is the work of Shah and Pantelides[104] that address the problem of design of multipurpose batch plants considering different schedules for different sets of production requirements using a scenario-based approach[28] and an approximate solution strategy. Ierapetritou and Pistikopoulos[86] presented a two stage stochastic programming formulation for the problem of batch plant design and operations under uncertainty. Straub and Grossmann[109] proposed the idea of stochastic flexibility index in order to evaluate the effect of uncertainty quantitatively. Bansal et al.[9] proposed a parametric programming framework for the flexibility analysis design of linear systems and in their subsequent work, that approach is generalized and unified to handle nonlinear systems. The multiperiod planning and scheduling of multiproduct plants under demand uncertainty is addressed by Petkov and Maranas[81]. The stochastic elements in their proposed model are expressed with equivalent deterministic forms, resulting in a convex mixed integer nonlinear programming (MINLP) problem. Schmidt and Grossmann[101] considered the optimal scheduling of new product testing tasks and reformulated the initial nonlinear, nonconvex disjunctive model as a MILP using different sets of simplifying assumptions that give rise to different models. The uncertainties in planning and scheduling problems are generally described through probabilistic models. During the last decade, fuzzy set theory has been applied to scheduling optimization using heuristic search techniques[36,53]. Recently, Balasubramanian and Grossmann[7] developed MILP models for flowshop scheduling and new product development processing scheduling, based on a fuzzy representation of uncertainty. Daniels and Carrillo[25] addressed the problem of β-robust scheduling in single-stage production facilities with uncertain processing times. Vin and Ierapetritou[111] proposed a multiperiod programming model to improve the schedule performance of batch plants under demand uncertainty. Acevedo and Pistikopoulos[1] addressed linear process engineering problems under uncertainty using a branch and bound algorithm, bas ed on solution of multiparametric linear programs at each node of the tree, and the evaluation of the uncertain parameters space for which a node must be considered.

However, most of the existing approaches can handle only a certain type of uncertain parameters, mostly uncertainty in product demands, and more importantly, the additional complexity constitutes them infeasible for realistic applications. An integrated framework is developed in this chapter in order to address the issue of uncertainty in short-term scheduling. The idea of inference-based sensitivity analysis for MILP problem is utilized within a branch and bound solution framework to determine the importance of different parameters and constraints and to provide a set of alternative schedules for the range of uncertain parameters under consideration.

This chapter is organized as follows. In the next section, the basic background of deterministic model adopted, parametric programming, and robustness metrics is provided, followed by the proposed approach for scheduling under uncertainty.

Section 6.4 is then utilized to present the results of two examples. Finally the work is summarized and the future work directions are discussed in section 6.5.

6.2 Background

6.2.1 Deterministic Scheduling Formulation

In this chapter, the mathematical model used for batch plant scheduling follows the main idea of continuous time formulation proposed by Ierapetritou and Floudas[51]. The model involves the following constraints:

$$minimize \quad H \quad or \quad maximize \quad \sum_s \sum_n price(s)d(s,n) \tag{6.1}$$

$$subject \quad to: \quad \sum_{i \in I_j} wv(i,j,n) \leq 1 \tag{6.2}$$

$$st(s,n) = st(s,n-1) - d(s,n)$$
$$+ \sum_{i \in I_j} \rho^p(s,i) \sum_{j \in J_i} b(i,j,n-1) + \sum_{i \in I_j} \rho^c(s,i) \sum_{j \in J_i} b(i,j,n) \tag{6.3}$$

$$st(s,n) \leq stmax(s) \tag{6.4}$$

$$Vmin(i,j)wv(i,j,n) \leq b(i,j,n) \leq Vmax(i,j)wv(i,j,n) \tag{6.5}$$

$$\sum_n d(s,n) \geq r(s) \tag{6.6}$$

$$Tf(i,j,n) = Ts(i,j,n) + \alpha(i,j)wv(i,j,n) + \beta(i,j)b(i,j,n) \tag{6.7}$$

$$Ts(i,j,n+1) \geq Tf(i,j,n) - U(1 - wv(i,j,n)) \tag{6.8}$$

$$Ts(i,j,n) \geq Tf(i^{'},j,n) - U(1 - wv(i^{'},j,n)) \tag{6.9}$$

$$Ts(i,j,n) \geq Tf(i^{'},j^{'},n) - U(1 - wv(i^{'},j^{'},n)) \tag{6.10}$$

$$Ts(i,j,n+1) \geq Ts(i,j,n) \tag{6.11}$$

$$Tf(i,j,n+1) \geq Tf(i,j,n) \tag{6.12}$$

$$Tf(i,j,n) \leq H \tag{6.13}$$

$$Ts(i,j,n) \leq H \tag{6.14}$$

where U denotes an upper bound of the makespan, for the cases where the objective is the minimization of makespan. For the cases where maximization of profit is considered, $U = H$ in constraints (6.8) to (6.10). In general, the objective function is to minimize the makespan as shown in (1) or to maximize the total profit. Allocation constraints (6.2) state that only one of the tasks can be performed in each unit at an event point (n). Constraints (6.3) represent that the material balances for each material at each event point (n) is equal to that at event point (n-1), adjusted by any amounts produced and consumed between event points (n-1) and (n), and delivered to the market at event point (n). The storage and capacity limitations of production units are expressed by constraints (6.4) and (6.5). Constraints (6.6) are written to satisfy the demands of final products. Constraints (6.7) to (6.14) represent time limitations due to task duration and sequence requirements in the same or different production units. Extensive testing of this model is reported in the literature[70] that compares favourably this model to other existing discrete and continuous time formulation for batch plant scheduling.

6.2.2 Parametric Programming

There is an increasing interest in the last few years for parametric programming mainly focusing on integer programming problem including linear and nonlinear formulations[31,32]. In this section a brief description of the basic approach proposed by Pertsinidis et al.[80] is described that covers the RHS scalar variations for MILP problem. This approach was selected for its simplicity in order to compare the results obtained from the proposed uncertainty analysis framework. The MILP problem considered has the following form:

$$
\begin{aligned}
z(\theta) = min \quad & c^T x + d^T y \\
s.t. \quad & Ax + Dy \le b(\theta) \\
& x^L \le x \le x^U \\
& \theta^L \le \theta \le \theta^U \\
& x \in R^m, y \in (0,1)^t
\end{aligned}
\tag{6.15}
$$

where $b(\theta) = b_0 + \theta r$ represents the vector of uncertain parameters, b_0 and r are constants, and scalar θ varies in the interval $[\theta^L, \theta^U]$. First the problem is solved at point $b = b_0 + \theta^L r$, and the optimal solution is obtained to be (x^*, y^*) with objective val ue z^L. The aim is then to identify a break point θ', beyond which the current integer solution is no longer optimal.

By performing a LP sensitivity analysis on the above formulation with fixed binary variables y^*, one can observe that the optimal value z varies with respect to θ as a linear function:

$$
z(\theta) = z^L + \lambda\theta, \quad \forall \theta^L \le \theta \le \theta^{U'} \le \theta^U \tag{6.16}
$$

where $[\theta^L,\ \theta^{U'}]$ is the sensitivity range of the current basis. By including the above equality, which should hold at the break point, and an integer cut to exclude the current optimal solution, the following MILP problem is solved to obtain θ' and a new solution (x'^*, y'^*).

$$\begin{aligned}
z(\theta) = min \quad & c^T x + d^T y \\
s.t. \qquad & Ax + Dy - \theta r \leq b_0 \\
\sum_{i \in F^1} y_i - & \sum_{i \in F^0} y_i \leq |F^1| - 1 \\
c^T x + d^T & y - z^L - \lambda\theta = 0 \\
& x^L \leq x \leq x^U \\
& \theta^L \leq \theta \leq \theta^{U'} \\
& x \in R^m, y \in (0,1)^t
\end{aligned} \tag{6.17}$$

where F^1 and F^0 denote the index sets of the binary variables with values equal to 1 and 0, respectively, at the current solution and $|F^1|$ is the cardinality of F^1. The next break point and the successor optimal schedule can be obtained through the above procedure within the interval $[\theta', \theta^U]$.

6.2.3 Robustness Metrics

In order to improve the schedule flexibility prior to its execution, it is important to measure the performance of a deterministic schedule under changing conditions due to uncertainty.

Standard Deviation (SD) is one of the most commonly used metrics to evaluate the robustness of a schedule. To evaluate the SD, the deterministic model with a fixed sequence of tasks $(wv(i,j,n))$ is solved for different realizations of uncertain parameters that define the set of scenarios k which results in different makespans H_k. The SD is then defined as:

$$SD_{avg} = \sqrt{\sum_k \frac{(H_k - H_{avg})^2}{(p_{tot} - 1)}}, \qquad H_{avg} = \frac{\sum_k H_k}{p_{tot}} \tag{6.18}$$

where H_{avg} is the average makespan over all the scenarios, and (p_{tot}) denotes the total number of scenarios. A detailed discussion of different robustness metrics can be found in[98].[111] proposed a robustness metric taking into consideration the infeasible scenarios. In case of infeasibility, the problem is solved to meet the maximum demand possible by incorporating slack variables in the demand constraints. Then the inventory of all raw materials and intermediates at the end of the schedule are used as initial conditions in a new problem with the same schedule to satisfy the unmet demand. The makespan under infeasibility (H_{corr}) is determin ed as the sum of those two makespans. Their proposed robustness metric is defined as:

$$SD_{corr} = \sqrt{\sum_k \frac{(H_{act} - H_{avg})^2}{(p_{tot} - 1)}} \tag{6.19}$$

where $H_{act} = H_k$, if scenario k is feasible and $H_{act} = H_{corr}$, if scenario k is infeasible.

6.3 Scheduling under Uncertainty

6.3.1 MILP Sensitivity Analysis

The formulation presented in section 6.2.1 corresponds to Mixed Integer Linear Programming (MILP) problem where the binary variables ($wv(i,j,n)$) denote the assignment of tasks (i) to units (j) at event point (n), respectively, throughout the time horizon. Therefore, the effects of operation parameters on the plant performance can be investigated through the sensitivity analysis of the MILP model of the deterministic scheduling problem. Although sensitivity analysis theory is well developed in linear programming, efforts are still being made in order to handle the integer programming case mainly due to lack of optimality criteria for the integer optimization problems. Schrage and Wolsey[102] examined the effect of a small perturbation on the right-hand side or objective function coefficients in an integer program by collecting dual information at each node of the branch and bound tree while solving the original integer program and using a recursive scheme to obtain an upper bound on the objective function. Their results are extended to non-linear integer programming problems by Skorin-Kapov and Granot[106]. An algebraic geometry algorithm for solving integer programming problems is presented by Bertsimas et al.[15] Their method provides a natural generalization of the Farkas lemma for IP and leads to a method of performing sensitivity analysis.

However, most of the existing approaches address the sensitivity analysis for pure 0-1 integer programming problems. A method of sensitivity analysis for mixed integer linear programming is presented by Dawande and Hooker[27] , based on the idea of inference duality. It reveals that any perturbation that satisfies a certain system of linear inequalities will reduce the optimal value no more than a prespecified amount. The inference-based sensitivity analysis consists of two parts: dual analysis that determines how much the problem can be perturbed while keeping the objective function value in a certain range, while primal analysis gives an upper bound on how much the objective function value will increase if the problem is perturbed by a certain amount. cuts. tions, the More specifically the main results of the inference-based sensitivity analysis are summarized below.

For the general mixed integer problem:

$$\begin{aligned} minimize \qquad & z = cx \\ subject \quad to \quad & Ax \geq a \\ 0 \leq x \leq h,\ x_j \ \ & integer, j = 1, ..., k \end{aligned} \tag{6.20}$$

Assuming a perturbation of all problem parameters such that:

$$\begin{aligned} minimize \qquad & z = (c + \Delta c)x \\ subject \quad to \quad & (A + \Delta A)x \geq a + \Delta a \\ 0 \leq x \leq h, x_j \ \ & integer, j = 1, ..., k \end{aligned} \tag{6.21}$$

If there exist $s_1^p, ..., s_n^p$ that satisfy the following set of inequalities, the constraint $z \geq z^* - \Delta z$ remains valid:

- for the perturbations ΔA and Δa in the parameters involved in the left and right hand side of the constraints:

$$\lambda_i^p \sum_{j=1}^{n} A_{ij}\underline{u}_j^p + \sum_{j=1}^{n} s_j^p(\overline{u}_j^p - \underline{u}_j^p) - \lambda_i^p \Delta a_i \leq r_p$$

$$r_p = -\sum_{j=1}^{n} q_j^p \overline{u}_j^p + \lambda^p a - z_p + \Delta z_p \tag{6.22}$$

$$s_j^p \geq \lambda_i^p \Delta A_{ij}, \quad s_j^p \geq -q_j^p, \; j = 1, ..., n$$

- for a perturbation Δc of the coefficients of the objective function:

$$\sum_{j=1}^{n} \Delta c_j \underline{u}_j^p - s_j^p(\overline{u}_j^p - \underline{u}_j^p) \geq -r_p \tag{6.23}$$

$$s_j^p \geq -\Delta c_j, s_j^p \geq -q_j^p, \; j = 1, ..., n$$

where $q_j^p = \lambda_i^p A_{ij} - \lambda_0^p c_j$; p corresponds to the leaf node where the dual variable of the objective function (λ_0^p) equals to 1, whereas $\underline{u}_j^p$ and $\overline{u}_j^p$ denote the lower and upper bound of x_j at node p, respectively; z_p is the objective value at node p; and $\Delta z_p = z^* - z_p$. Thus by utilizing constraints (6.22), (6.23) in the scheduling problem, the range of parameters where the objective remains within certain limits can be identified and used to evaluate alternative schedules at the branch and bound tree. Morever the import ance of different constraints and parameters are determined and can be utilized to improve future plant operability.

6.3.2 Proposed Uncertainty Analysis Approach

The basic idea of the proposed approach is to utilize the information obtained from the sensitivity analysis of the deterministic solution to determine (a) the importance of different parameters and constraints and (b) the range of parameters where the optimal solution remains unchanged. The main steps of the proposed approach are shown in Figure 6.1.

More specifically, there are two parts in the proposed analysis. In the first part, important information about the effect of different parameters is extracted following

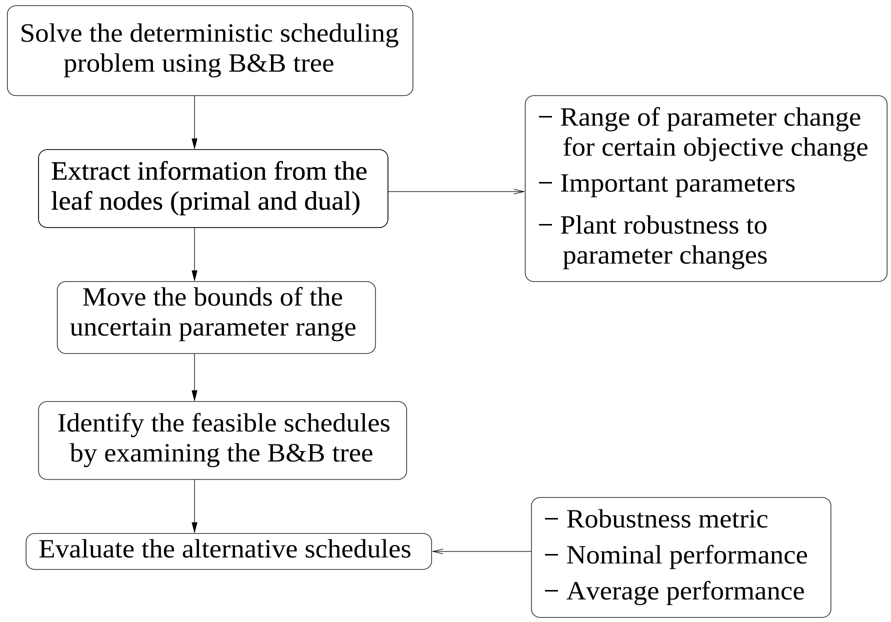


Figure 6.1: Flow Chart of Proposed Approach

the sensitivity analysis step, whereas in the second part alternative schedules are determined and evaluated for different uncertainty ranges.

First, the deterministic scheduling is solved at the nominal values using a branch and bound solution approach, and the dual multipliers $\lambda^{\mathbf{p}}$ are collected at each leaf node p. Then the inference-based sensitivity analysis as described in previous section is performed for all the important scheduling parameters, including demands, prices, processing times and capacities. Note that, only the dual information of the nodes that correspond to nonzero dual variables is required. Using the results of this analysis, a number of very interesting questions can be answered regarding the plant robustness to parameter changes.

In particular:

- how the capacity of the units affect the production objective?

- what is the range of product demand that can be covered and how much the profit would be affected by such changes?

- what is the effect of price change at the objective value?

- what is the significance of the constraints involved in the model? Are there any redundant set of constraints?

The first question can be answered by imposing the same perturbation to capacity constraints (6.5) for the different units involved in the production of specific products and determine the change in the objective function (Δz). The unit with the largest effect in the objective value is also the most critical one for the production of this product and thus a change in its capacity will result in the largest production change. Similarly, the rest of the questions can be answered by analyzing perturbations at the appropriate constraints together with the effects in the objective function. Results for two examples are given in the next section.

In the second part of the analysis, the sensitivity information is used to define the range of uncertain parameters where the schedule is optimal and identify alternative schedules at different uncertainty ranges. The set of constraints (6.22), (6.23) are used to determine the range of uncertain parameters for certain changes in the objective function. The branch and bound procedure is then continued on the nodes with the objective value within the predicted limits to identify new optimal solutions. The alternative schedules are evaluated using the robustness metric (SD_{corr}) as defined in section 6.2.3, the average and the nominal schedule performance in terms of the objective function.

Since the entire analysis is based on a single among a large number of possible branch and bound trees that can be used to solve the MILP , it provides conservative sensitivity ranges. Theoretically, the exact sensitivity ranges can be obtained by investigating an exponential number of branch and bound trees. However, using the above analysis, useful information can be extracted regarding the approximate range of parameter change for certain ob jective change, plant robustness to parameter changes, as well as to determine the importance of different parameters as illustrated in the next section.

Figure 6.2: State-Task Network Representation for Example 1

6.4 Illustrative Examples

Example 1 involves a single product production line consisting of a mixer, a reactor and a purifier as shown in Figure 6.2. The data for this example are obtained from [51]. First, the problem is solved with the objective of maximizing the production within the time horizon of 8 hours, and the dual informaton (λ^p) is collected at each leaf node. The pertubations of capacities, demands and prices can be viewed as ΔA, Δa and Δc in constraint sets (6.22) and (6.23), while the production change corresponds to Δz. For a given amount of production change Δz, one of these tw o linear systems (depending on which parameter is examined) is solved at each leaf node. By checking if there exist $s_1, s_2, ..., s_n$ that satis fy the linear system at each leaf node, we can determine the maximum allowable perturbation. The following important findings are revealed. First, by perturbing the capacity constraints, it is found that the most critical unit of the production line is the purifier. By increasing its capacity, production will increase accordingly. The same information can be used to determine the allowable purifier size reduction that will result in limiting production decrease. For example it is found that the purifier can decrease by up to 4.24 units without reducing the profit by more than 5%. Important sensitivity information can be obtained by a simple primal analysis of the MILP problem that results in an upper bound on the optimal value given a perturbation of the problem data. In this way it is determined that a 10% increase in purifier size from 50 to 55 units, will result in a maximum of 2.75% increase in production profit from 71.16 to 73.12 units (upper bound on the objective function). The results of the sensitivity analysis can be also used to analyze the effects of product price in the objective function in order to investigate the relative importance of different products as well as to make decisions regarding the production levels. For this example it is found that there is one to one correspondence between price increase and objective function increase. In particular, a 5% price increase would result in also 5% increase in the objective function. The question of how demand fluctuations affect the profit function and the production time is also analyzed. In order to consider this factor, the required dual information involves the dual variables associated with the demand constraint at the nodes that this constraint is active. It is found that a demand change from 60 to 65 units (8.3%) can be accommodated within the production line resulting in an equivalent increase of production profit but also in an increased, by 2.6%, makespan, 11.59 hours from 11.29 hours.

For the second part of the analysis, first the demand is considered to be the uncertain parameter varying within the range of $[20, 100]$. The problem is solved at nominal demand (50) with objective function of minimum makespan. A branch and bound tree is constructed as illustrated in Figure 6.3, where the value next to each node indicates the objective value of relaxed LP and the binary variables that are branched at each level are shown at the right side. Applying the inference duality sensitivity analysis, the following expression is obtained regarding the range of demand change following specific objective change (ΔH): $-0.0966\Delta d \leq \Delta H$, which means that if the demand is increased by Δd, the new makespan becomes at most $H_{nom} + 0.0966\Delta d$.

When the demand is increased to 80, the current schedule becomes infeasible. Using the previous inequality, we get $H' \leq 12.73$. By solving the LP problem at the leaf nodes with the perturbed demand, the remaining nodes are nodes 1 and 2, with objective values 12.13 and 12.73, respectively. Therefore, we continue the branch and bound procedure on these two nodes and find the optimal schedule 2 and a feasible schedule 3. After the schedules are determined to cover the uncertainty range under consideration, the schedules are evaluated based on the corrected SD robustness metric defined in equation (6.34) considering 5 scenarios in the demand interval $[20, 100]$. The average makespan, standard deviation and the nominal makespan are shown in Table 6.1. Note that although schedule 1 provides the best solution at nominal point, it has poor performance over the entire interval. Schedule 2 apparently has the

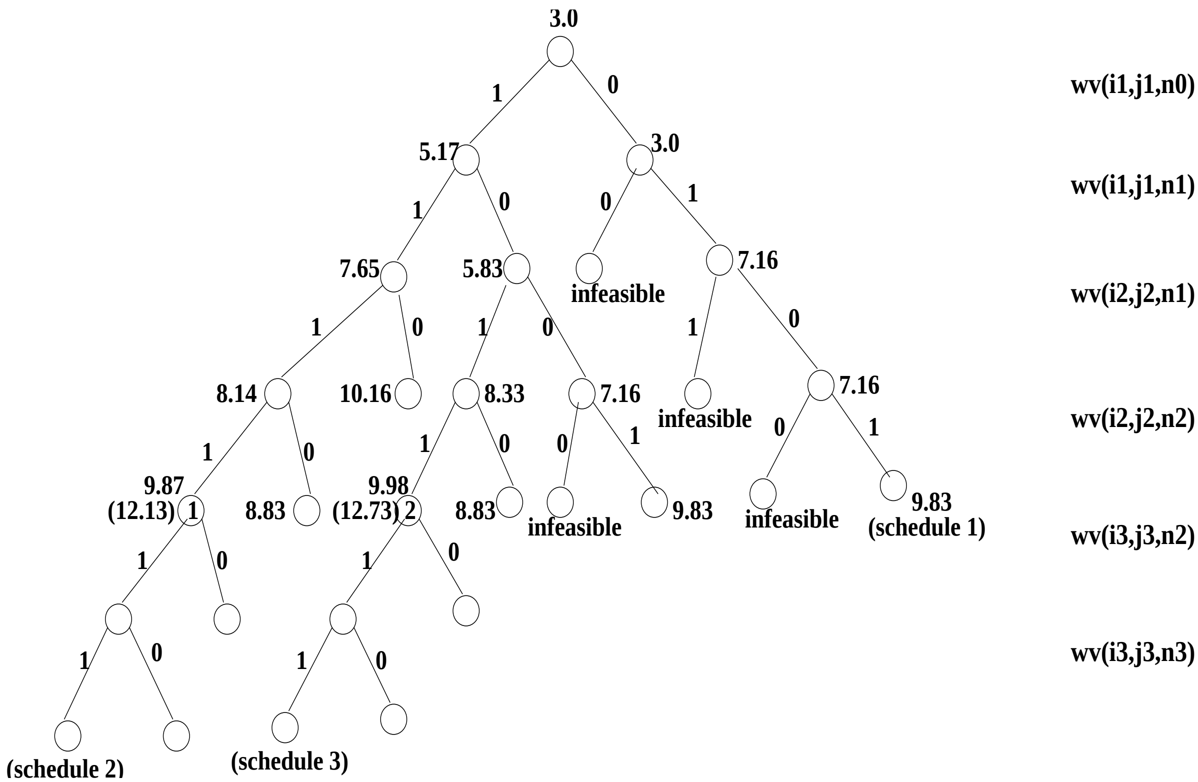


Figure 6.3: Branch and Bound Tree for Example 1 with Uncertain Demand

highest robustness and provides the shortest average makespan. Schedule 3 provides a compromise solution with robustness between the ones of schedules 1 and 2 and average and nominal performance close to the best ones given by schedule 2.

	Schedule 1	Schedule 2	Schedule 3
H_{nom}(h)	9.83	10.77	10.91
H_{avg} (h)	14.20	11.56	11.79
SD_{corr}	5.52	1.61	2.17

Table 6.1 Comparison of Alternative Schedules for Example 1 Considering Uncertain Demand

Next the processing time of mixing is considered as uncertain parameter, which is denoted as $\alpha(i,j)$ in the deterministic model. The branch and bound tree is built as depicted in Figure 6.4 at the nominal point ($\alpha(i1,j1) = 3.0$) with the objective of maximizing profit and schedule 1 is obtained as the optimal solution. Note that the negative values appear since in order to apply the dual analysis, the maximization objective is transformed to a minimization problem. When $\alpha(i1, j1)$ is increased to 4.0, the current schedule becomes infeasible. The new objective value is estimated to be $-Profit' \leq -Profit_{nom} + 24.48\Delta\alpha = -47.04$. By investigating the nodes that follow nodes 1, 2 and 3, the optimal solut ion is found to be the equivalent schedules 2 and 4, while schedule 3 is a feasible solution. Equivalent schedules are schedules that have different values for binary variable sets but correspond to the same scheduling sequence. This is due to model redundancy that enables the determination of a variety of scheduling alternatives. Those schedules are then evaluated with respect to their robustness within the region of uncertainty range of $\alpha(i1, j1)$ $[3.0, 4.0]$ and the results are shown in Table 6.2. The optimal schedule at nominal point has the

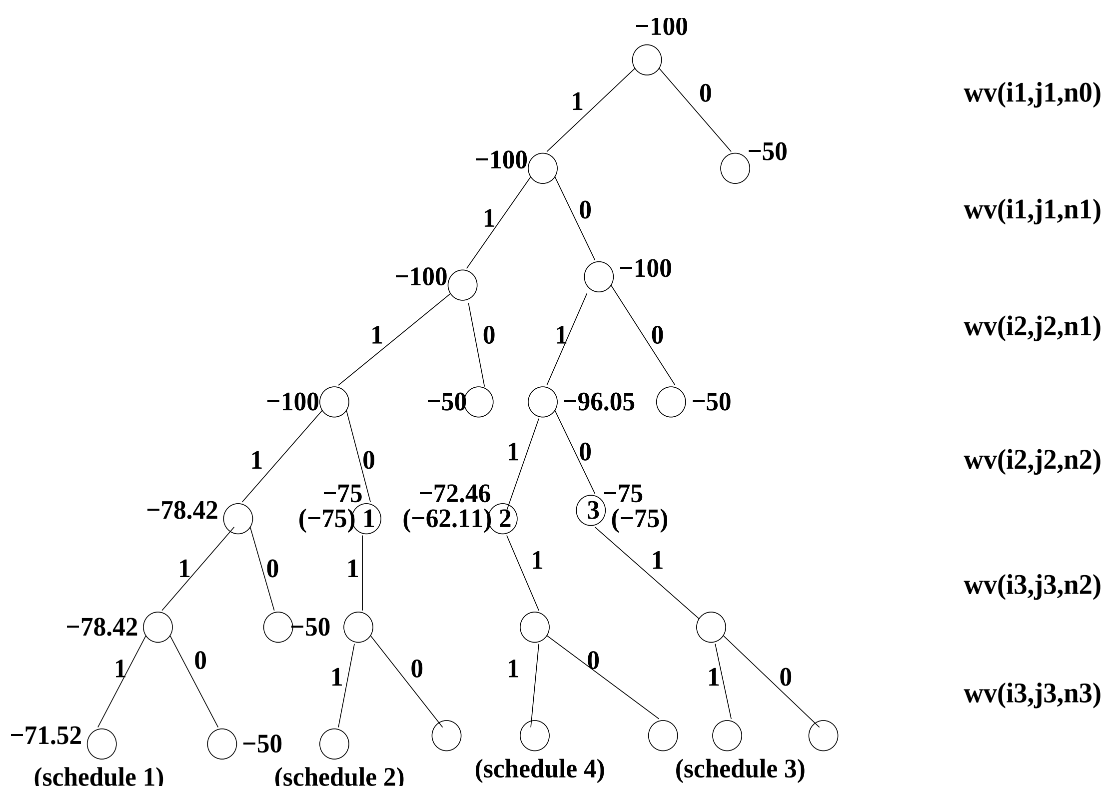


Figure 6.4: Branch and Bound Tree for Example 1 with Uncertain Processing Time

largest SD, while schedule 2 gives the best solution robustness. Schedule 3 provides a better average profit compared to schedule 2, therefore it provides an attractive alternative.

Example 2[51] considers two different products produced through five processing

	Schedule 1	Schedule 2	Schedule 3
$-\text{Profit}_{nom}$	−71.52	−65.27	−65.27
$-\text{Profit}_{avg}$	−66.98	−64.61	−65.17
SD_{corr}	26.9	9.33	10.49

Table 6.2 Comparison of Alternative Schedules for Example 1 Considering Uncertain Processing Time

stages: heating, reactions 1, 2 and 3, and separation of product 2 from impure E as illustrated in the state-task network (STN) representation in Figure 6.5. For the first part of the analysis, the problem is solved with the objective of maximizing the profit within the time horizon of 12 hours. After the sensitivity analysis is performed, the following information is obtained. It is found that the most critical task of the production line is reaction 2. By decreasing the processing capacity of reaction 2 in reactor 1 or reactor 2 by 11 units, the profit will be reduced by 5%, whereas very small change or no change at all is observed at the objective function with reaction 1 or reaction 3 processing capacity change in both reactors. The objective value is also not sensitive to the change of other parameters, for example, the processing capacity of separation in separator can drop by up to 120 units without decreasing the profit. Another important modeling issue that can be addressed is the question of constraint redundancy. Here the importance of storage constraints are investigated and it is found that these constraints are redundant since they are not active in any of the solution branch and bound nodes. More interestingly, the duration constraints are also found to be redundant, which means that the maximum processing capacities are already reached with the current processing times, so that the profit cannot be improved even with zero processing times assuming fixed number of event points. For the second part of the analysis, the demand of product 2 is considered to be the uncertain parameter varying within the range of $[20, 80]$ and the objective function is modified to minimize the makespan. A branch and bound tree is constructed at nominal point $r('p2') = 50$ and the dual information is stored at each node. Applying the inference duality sensitivity analysis, the following expression is

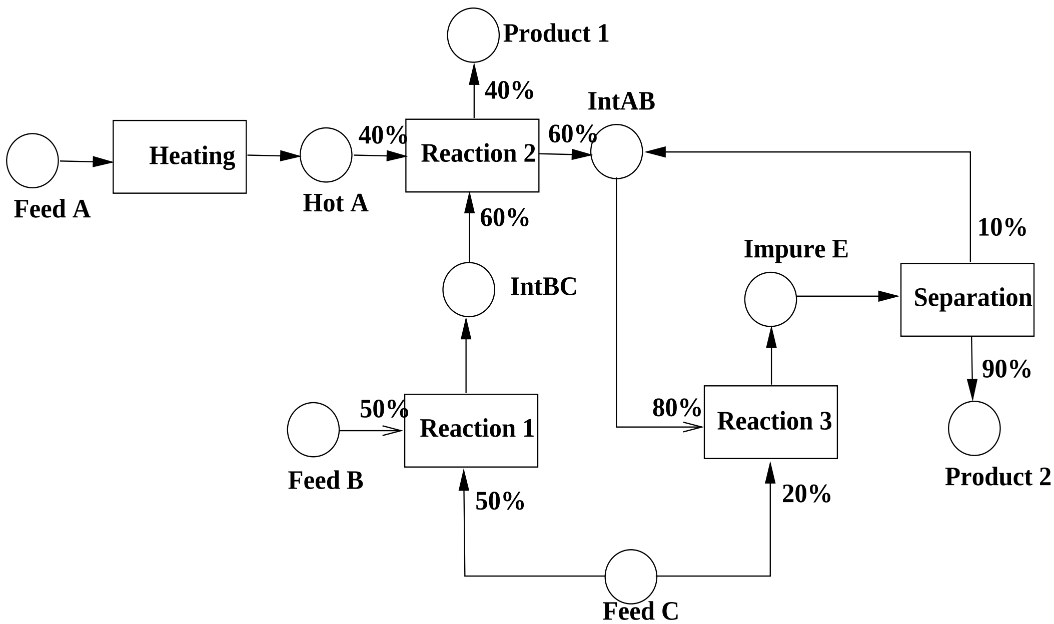


Figure 6.5: State-Task Network Representation for Example 2

	Schedule 1	Schedule 2	Schedule 3
H_{nom}(h)	7.00	7.14	7.40
H_{avg} (h)	8.15	7.24	7.40
SD_{corr}	2.63	0.29	0.27

Table 6.3 Comparison of Alternative Schedules for Example 2

obtained regarding the range of demand change following specific objective change (ΔH): $-0.0297\Delta d \leq \Delta H$, which means that if the demand is increased by Δd, the new makespan becomes at most $H_{nom} + 0.0297\Delta d$. When $r('p2')$ is increased from 50 to 80, schedule 1 becomes infeasible. Then, we solve the LP problem at each leaf node with the demand of 80 and check the leaf nodes with the objective value below 7.89 that is obtained using this inequlaity in the branch and bound tree. The new optimal solution is found to be schedule 2 and schedule 3 is one feasible solution. The schedules are then evaluated with respect to the mean and nominal makespan and standard deviation within the demand range of $[20, 80]$ and the values are shown in Table 6.3. Comparing with schedule 2, schedule 3 has larger mean makespan but lower standard deviation, which means higher robustness, therefore de pending on the decision maker's attitude towards risk and the expected demand growth one can choose schedule 3 compared to schedule 2, whereas schedule 1 remains a valid alternative if the demand is expected to remain constant.

Note that although example 2 is bigger and more complex than example 1, the proposed uncertainty analysis does not substantially increase the problem complexity. That is due to the fact that the required information is already obtained from the solution of the deterministic problem.

	(Schedule 1)				(Schedule 2)				(Schedule 3)			
(task, unit)	n0	n1	n2	n3	n0	n1	n2	n3	n0	n1	n2	n3
(heating, heater)	1	0	0	0	1	1	0	0	1	0	0	0
(reaction 1, reactor 1)	0	0	0	0	1	0	0	0	1	0	0	0
(reaction 1, reactor 2)	1	1	0	0	1	0	0	0	1	0	0	0
(reaction 2, reactor 1)	0	1	0	1	0	1	0	1	0	1	0	0
(reaction 2, reactor 2)	0	0	0	0	0	1	0	0	0	1	0	0
(reaction 3, reactor 1)	0	0	1	0	0	0	1	0	0	0	1	0
(reaction 3, reactor 2)	0	0	1	0	0	0	1	0	0	0	1	0
(seperation, still)	0	0	0	1	0	0	0	1	0	0	0	1

Table 6.4 Values of Binary Variables of Optimal Schedules for Example 2

6.5 Summary and Future Work

An integrated framework is developed in this chapter to handle uncertainty in short-term scheduling based on the idea of inference-based sensitivity analysis for MILP problem and the utilization of a branch and bound solution methodology. The proposed method leads to the determination of the importance of different parameters and the constraints to the objective function and the generation and evaluation of a set of alternative schedules given the variability of the uncertain parameters. Two illustrative examples are presented to highlight the information extracted by the proposed approach and the complexity involved.

The main advantage of the proposed method is that no substantial complexity is added compared with the solution of the deterministic case since the only additional required information is the dual information at the leaf nodes of the branch and bound tree. However, at the current step, the retrieval and storage of dual information of the leaf nodes are processed manually. That is apparently prohibitive when solving large-scale problem. It is our future direction to automatically extract and store the required dual information of the leaf nodes in the branch and bound tree. This will enable the utilization of the proposed approach in large scale problems and the consideration of multiple parameters. As mentioned in section 6.3.2, the entire analysis is based on a single branch and bound tree used to solve the MILP out of an exponential number of alternatives. Hence one future work direction is to find a way of investigating a large number of branch and bound trees so as to obtain more accurate sensitivity range. Another future direction is to define better rules when moving along the range of uncertain parameters in order to find alternative schedules.

Nomenclature

Indices

i = tasks

j = units

k = scenarios

n = event points

s = states

Parameters

p^k = probability of scenario k

price(s) = price of state s

$\rho^p(s,i), \rho^c(s,i)$ = proportion of state s produced, consumed from task i, respectively

r(s) = market requirement for state s at the end of the time horizon

stmax(s) = maximum storage capacity of state s

Vmin(i,j) = minimum capacity of unit j when processing task i

Vmax(i,j) = maximum capacity of unit j when processing task i

Variables

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

H = time horizon

st(s,n) = amount of state s at event point n

Ts(i,j,n) = starting time of task i in unit j at event point n

Tf(i,j,n) = finishing time of task i in unit j at event point n

wv(i,j,n) = binary variable that assigns task i to unit j at event point n

Chapter 7

Uncertainty Analysis on the Right-Hand-Side for MILP Problems

7.1 Introduction

A number of problems from the area of process design and operations are commonly formulated as mixed integer linear programming (MILP) problems. One way to incorporate uncertainty into these problems is using MILP sensitivity analysis and parametric programming methods. The main limitation of most existing methods is that they can only be applied to problems with a single uncertain parameter or several uncertain parameters varying in a single direction. A number of approaches have been developed for parametric integer programming problems that involve a single parameter/scalar variation, basically including implicit enumeration methods (Roodman[92], Piper and Zoltners[85]), branch and bound methods (Roodman[93], Marsten and Morin [71], Ohtake and Nishida[77]), and cutting plane methods (Holm and Klein[49], Jenkins and Peters[56]), etc. A detailed literature review can be found in Jenkins[55].

Jenkins' approach is extended by Crema[24] for the multiparametric 0-1 integer linear programming (ILP) problem considering the perturbation of the constraint matrix, the objective function and the RHS vector. The proposed algorithm iteratively solves a nonlinear problem, which can be converted to an equivalent MILP formulation, in order to obtain a complete multiparametrical analysis.

Acevedo and Pistikopoulos[1] proposed a parametric programming approach for the analysis of linear process engineering problems under uncertainty. The procedure solves the multiparametric linear programming (mpLP) at each node of the B&B tree, then compares and identifies the different optimal integer solutions and their corresponding optimal value functions. Pertsinidis et al.[80] developed an algorithm for MILP sensitivity analysis. At each iteration, the LP sensitivity analysis results and a cut that excludes the current integer solution are incorporated to a MILP problem so as to find the breaking point and the successor optimal integer solution. Their ideas were extended by Dua and Pistikopoulos[32], by decomposing the mp-MILP into two subproblems and then iterating between them. The first subproblem is obtained by fixing the integer variables, resulting in a mpLP problem, whereas the second subproblem is obtained by relaxing the parameters as variables, leading to a MILP problem.

The problem of RHS multiparametric linear problem was first addressed by Gal and Nedoma[38]. Their algorithm is based on the Simplex algorithm for deteministic LPs. It starts with an initial optimal basis at a feasible point and moves to each of its possible neighbor bases by one dual step to determine the new optimal solution. This procedure is repeated until there is no optimal basis that still has unexamined neighbors. A geometric approach is proposed by Borrelli et al.[20], which is based on the direct exploration of the parameter space and their definition of critical regions is not associated with bases but with the set of active constraints.

In this chapter, we focus on the parametric MILP problems with RHS uncertain parameters which are allowed to vary independently. The proposed solution procedure starts with the B&B tree of the MILP problem at the nominal values of the uncertain parameters and requires two iterative steps: LP/mpLP sensitivity analysis and updating the B&B tree. Rather than checking all the neighbor bases of the associated LP tableau, a novel algorithm is developed for mpLP problems that can determine the optimal functions and their corresponding critical regions without constructing the LP tableaus.

This chapter is organized as follows. The details of the proposed approach for the cases of single and multiple uncertain parameters are presented in the next section. A number of case studies are provided in section 7.3 to illustrate the steps of the proposed approach. The work is summarized in section 7.4 and some of the ideas for future developments are discussed.

7.2 Proposed Framework

For the general mixed integer problem:

$$\begin{aligned} \textbf{(P1)} \qquad \min \qquad & z = cx \\ \text{subject to} \quad & Ax \geq \theta \\ x \geq 0,\ x_j \ \ & integer, j = 1, ..., k \end{aligned}$$

Assuming a perturbation of problem RHS parameter values such that:

$$Ax \geq \theta + \Delta\theta$$

The aim of this work is to investigate the effect of $\Delta\theta$ on the optimal solution x and objective value z. First the original problem is solved at a starting point following a Branch and Bound solution procedure, and the dual information λ^p, z^p is collected at each leaf node. Then LP sensitivity analysis is performed for the relaxed LP problems at the leaf nodes. The LP sensitivity analysis procedure for multiple uncertain parameters is more complicated than that of the single uncertain parameter, therefore, the detailed steps for these two cases will be presented separately in this section. When only one parameter is involved, the objective varies with the uncertain parameter in one direction. In order to identify the next optimal function, we can slightly increase the value of the uncertain parameter beyond the current optimal basis. The new dual information can then be found so as to determine the next optimal integer solution and update the B&B tree. However, when the objective value is a function of multiple parameters, it could be difficult to find a way to identify all the neighboring bases. Therefore, a new algorithm is developed as will be presented later. After the mpLP sensitivity analysis at the leaf nodes, a comparison procedure is required in order to update the B&B tree. The optimal functions at the leaf nodes are compared with the current upper bound in the same critical region. If an integer leaf node can provide a better objective then the upper bound is updated in that region. If a noninteger node provides a better objective then it should be branched. The mpLP is performed at the new leaf nodes and the comparison procedure continues. This procedure stops when no further branching is required.

7.2.1 Single Uncertain Parameter

For the case of single uncertain parameter, the proposed approach follows the basic ideas of the interactive reference point approach proposed by Alves and Climaco[5] presented for multiple objective MILP problems. The proposed framework is shown in Figure 7.1.

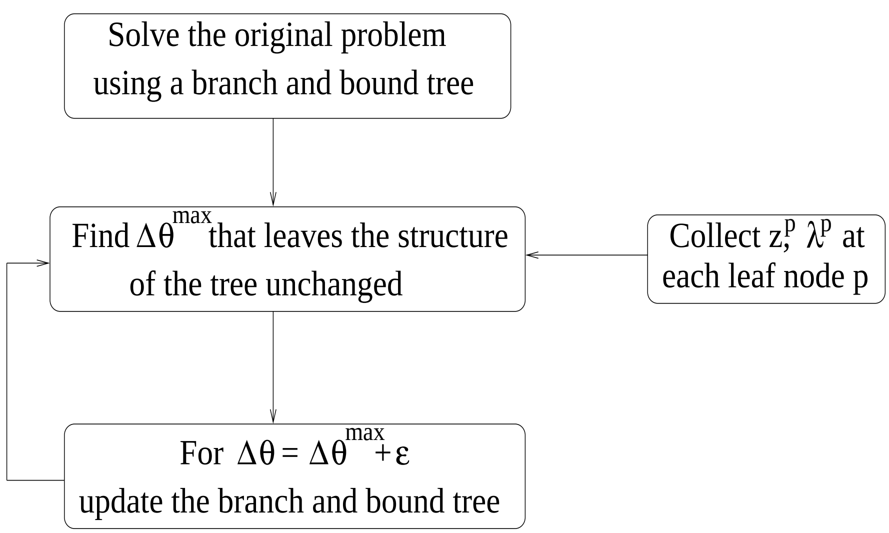


Figure 7.1: Flow Chart of Proposed Approach for Single Uncertain Parameter Case

First the problem is solved at the nominal values of the uncertain parameters using a branch and bound solution approach, and the dual information λ^p, z^p is collected at each leaf node. Assuming that the optimal solution is found at node 0, the LP sensitivity analysis is then performed at node 0 to determine the range $\Delta\theta^{basis}$ within which the current optimal basis does not change. We need to find the perturbation $\Delta\theta^{max}$ beyond which the structure of the branch and bound may not remain the same. $\Delta\theta^{max}$ can be found through the following equation:

$$\Delta\theta^{max} = min\{\Delta\theta^{basis}, min\{\frac{z^p - z^0}{\lambda^0 - \lambda^p}\}\}$$

Where z^0 and λ^0 are the objective value and dual multiplier at the optimal node 0, respectively. Note that only the positive $\frac{z^p - z^0}{\lambda^0 - \lambda^p}$ need to be considered, because the negative one means that node p can never provide a better solution than node 0 at a certain point.

The next step is to update the tree when $\Delta\theta = \Delta\theta^{max} + \epsilon$ is slightly larger than $\Delta\theta^{max}$. The LP problems of the leaf nodes are solved at the new θ value and the branching procedure continues so as to determine the new optimal node. The following three cases can be observed:

Case 1: The optimal node yields noninteger solution and the new optimal solution is provided by its descent node.

Case 2: The new optimal node is a descent node of other leaf node.

Case 3: The optimal node 0 still provides the optimal solution, but the basis has changed.

The procedure continues by determining the new $\Delta\theta^{max}$, and these steps are executed repeatedly until no feasible solution exists beyond the current range.

7.2.2 Multiple Uncertain Parameters

This subsection presents the detailed steps (Figure 7.2) of the proposed approach to deal with the case of multiple uncertain parameters. Assuming for simplicity in the presentation that we want to investigate two parameters, θ_a and θ_b, changing in the

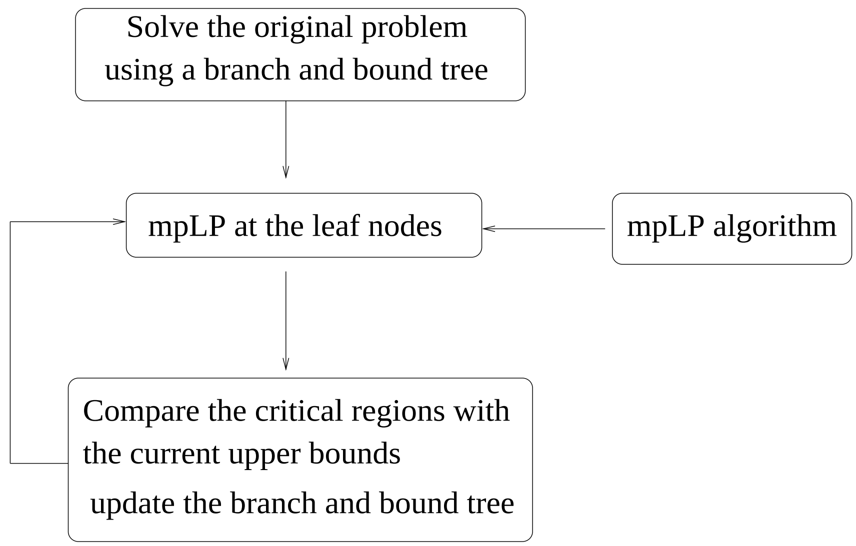


Figure 7.2: Flow Chart of Proposed Approach for Multiple Uncertain Parameters Case

range of $[a_0, a_0 + \Delta a]$ and $[b_0, b_0 + \Delta b]$. The MILP problem is first solved at (a_0, b_0) using branch and bound algorithm and the optimal solution is found at node 1 (Figure 7.3). Other leaf nodes of the B&B tree are denoted as node 2, node 3, ..., node n. Note that only the information at the leaf nodes is required. This is true since if at another set of uncertain parameters (a', b'), there exists a new optimal solution, it can always be uncovered by checking or continuing the branching procedure on the current leaf nodes. Let's assume for example that the new optimal solution can be provided by a non-leaf node (node A). With the original data, the relaxed LP problem of node A must have a partial integer solution, otherwise it is a leaf node. With the perturbed data, the LP problem of node A gives the optimal integer solution. According to our proposed method, all the current leaf nodes are examined that include the subsequent nodes of node A (node 2 and 3). Apparently, if node A yields an integer solution, either node 2 or 3 should provide that solution too. Thus, it is true that only the leaf nodes need to be examined at each iteration.

The next step in the proposed approach is to solve the multiparametric linear programming at each of the leaf nodes including node 1 (Figure 7.3), so as to identify the optimal value functions and their corresponding critical regions in the region

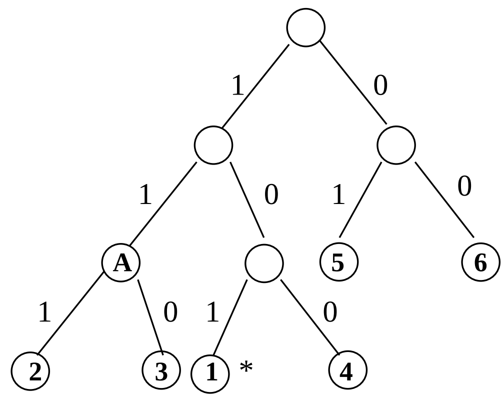


Figure 7.3: Branch and Bound Tree for the Original Problem

of interest. In this work, a new algorithm is proposed for the solution of mpLP. When the mpLP procedure is completed, the ouput is a set of optimal functions $z = z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b, k = 1, ...K$, where K is the number of critical regions. For any point (θ_a, θ_b) in the range of $[a_0, a_0 + \Delta a]$ and $[b_0, b_0 + \Delta b]$, the objective value cx^* of the relaxed LP problem of that node can be expressed by $\max\{z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b, k = 1, ...K\}$. If the procedure is not complete, then there must exist a point (θ_a, θ_b), such that $\max\{z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b, k = 1, ...K\}$ is less than cx^*. Thus a bilevel programming problem is formulated as follows:

$$
\begin{aligned}
\textbf{(P2)} \qquad & \max \qquad \{\min \; cx | Ax \geq \theta\} - z \\
\text{subject to} \quad & z \geq z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b, k = 1, ...K \\
& a_0 \leq \theta_a \leq a_0 + \Delta a \\
& b_0 \leq \theta_b \leq b_0 + \Delta b
\end{aligned}
$$

It is proved that linear bilevel programming problems (BLPP) are NP-hard[10]. In order to avoid solving a BLPP, we propose to first convert the relaxed LP problems at the leaf nodes to its dual form, so that the uncertain parameters appear in the objective function.

$$\begin{aligned} \textbf{(P3)} \qquad & \max \qquad \theta y \\ \text{subject to} \quad & A^T y \le c \\ & a_0 \le \theta_a \le a_0 + \Delta a \\ & b_0 \le \theta_b \le b_0 + \Delta b \\ & y \ge 0 \end{aligned}$$

Replacing the inside optimization problem of (P2) by its dual (P3), the problem can be solved as a single optimization problem (P4).

$$\begin{aligned} \textbf{(P4)} \qquad & \max \qquad \theta y - z \\ \text{subject to} \quad & A^T y \le c \\ z \ge z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b, & k = 1, ...K \\ & a_0 \le \theta_a \le a_0 + \Delta a \\ & b_0 \le \theta_b \le b_0 + \Delta b \\ & y \le 0 \end{aligned}$$

Starting from the nominal point (a_0, b_0), an optimal function is obtained with respect to the uncertain parameters as follows:

$z = z^{(1)} + \lambda^{(1)}\theta_a + \beta^{(1)}\theta_b$

and incorporated in problem (P4). In problem (P4), the objective function is to maximize the gap between the optimal objective value θy at any point (a, b) in the uncertain range and the maximum value of the optimal function, which is $max(z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b)$, and therefore takes the form $(\theta y - z)$. The constraints contain the original constraints and the current optimal functions $z = z^{(k)} + \lambda^{(k)}\theta_a + \beta^{(k)}\theta_b$, hence all the constraints of problem (P4) are linear. If the objective value of the (P4) model is nonzero, it means that there exists at least one point (θ_a, θ_b) at which its real objective value cannot be represented by any of the current objective value functions.

Therefore, the objective value function at that point is $z = z' + \lambda^{'}\theta_a + \beta^{'}\theta_b$ and should be included in the next iterations. At each iteration, (P4) problem is solved in order to identify any uncovered region. This procedure terminates when the objective value for problem (P4) is 0, which means that the entire uncertain parameter range is covered by the existing objective functions.

Note that since problem (P4) is a nonconvex problem due to the bilinear term in the objective function, it is solved using a global optimization solver GAMS/BARON [95], which relies on *branch-and-reduce* algorithm.

Therefore, by performing mpLP at each leaf node p, a number of critical regions (K), $CR_p^{(1)}$, $CR_p^{(2)}$, ..., $CR_p^{(K)}$ are identified and in each $CR_p^{(k)}, k = 1, ..., K$, the optimal value $z_p^{*(k)}$ is expressed as $z_p^{*(k)} = z_p^{(k)} + \lambda_p^{(k)}\theta_a + \beta_p^{(k)}\theta_b$.

Similar to the procedure presented for the single uncertain parameter case, the next step is to update the B&B tree. However, this step becomes more complicated since the parameters vary in multiple directions. The main procedure is to compare the critical regions of the leaf nodes with the current upper bounds and finally identify a set of new critical regions, and their corresponding objective function values and optimal integer solutions. At the beginning, the upper bounds CR^{UB} are set to be the critical regions of the current optimal node (node 1), which are $CR_1^{(1)}$, $CR_1^{(2)}$, ..., $CR_1^{(K)}$. Assuming that we want to compare critical regions CR_1^{UB} and $CR_2^{(2)}$, which have intersection CR^{int}, the following constraint is defined:

$$z_1^{UB} \geq z_2^{*(2)} \tag{7.1}$$

and a redundancy test for this constraint is solved in CR^{int} as follows[1]:

$$\begin{aligned} \textbf{(P5)} \qquad & \min \quad \epsilon \\ \text{subject to} \quad & z_1^{UB} = z_2^{*(2)} + \epsilon \\ z_1^{UB} &= z_1^{UB} + \lambda_1^{UB}\theta_a + \beta_1^{UB}\theta_b \\ z_2^{(2)} &= z_2^{(2)} + \lambda_2^{(2)}\theta_a + \beta_2^{(2)}\theta_b \\ & \theta_a, \theta_b \in CR^{int} \end{aligned}$$

The solution of problem (P5) can result into the following cases:

Case 1: Problem (P5) is infeasible. This implies that z_1^{UB} is smaller in CR^{int} and since it is a minimization problem, the solution of CR_1^{UB} and $z_1^{UB} = z_1^{UB} + \lambda_1^{UB}\theta_a + \beta_1^{UB}\theta_b$ remains to be the upper bound for CR^{int}, and node 2 will not provide a better integer solution in CR^{int} and thus fanthomed for CR^{int}.

Case 2: The solution of problem (P5) is positive and therefore constraint (1) is redundant. This means that z_1^{UB} is larger and therefore, the solution of $CR_2^{(2)}$ and $z_2^{*(2)} = z_2^{(2)} + \lambda_2^{(2)}\theta_a + \beta_2^{(2)}\theta_b$ are stored for CR^{int}. If node 2 represents an integer solution, then the optimal function for CR^{int} is updated to be $z_2^{*(2)}$. If node 2 does not correspond to an integer solution, then the information of this node should be stored and continue the branching procedure on node 2 in CR^{int} to identify new nodes.

Case 3: The solution of problem (P5) is negative, which means that constraint (1) is not redundant and thus CR^{int} is divided into two parts by constraint $z_1^{UB} \geq z_2^{*(2)}$, z_1^{UB}provides a better solution on one side in which the current upper bound doesn't change, whereas $z_2^{*(2)}$ gives a better solution on the other side, in which $z_2^{*(2)} = z_2^{(2)} + \lambda_2^{(2)}\theta_a + \beta_2^{(2)}\theta_b$ should be stored if node 2 is an integer node, otherwise node 2 will be branched to identify new nodes.

At each iteration, the new leaf nodes in the updated B&B tree will be compared to the current upper bounds, so as to determine the new optimal functions in their intersected region. This procedure stops when no further branching is required and the parametric solution at the uncertain range of interest can be represented by a number of critical regions that contain their corresponding optimal functions and integer solutions.

Comparing to the existing approach[1], the proposed method solves the mpLP at only the leaf nodes in the B&B tree instead of every node during the branch and bound procedure, and consequently reduces the computational efforts significantly as will be shown in the next section through a number of example problems. Moreover, the new mpLP approach can efficiently determine the optimal functions and critical regions without retrieving the LP tableaus and visiting the neighbor bases, which can be easily extended to large number of parameters.

7.3 Case Studies

In this section, a number of examples are presented for the cases of single and multiple uncertain parameters so as to clarify the methodologies described in section 7.2.

7.3.1 Examples for Single Uncertain Parameter Case

Example 1

$$\min \quad z = 2x_1 + 3x_2 + 1.5x_3 + 2x_4 + 0.5x_5$$

$$\text{subject to} \quad 2x_1 + x_2 + x_3 \geq 7 \tag{7.2}$$

$$2x_2 + x_4 + x_5 \geq 4 \tag{7.3}$$

$$x_3 + x_4 - x_5 \geq 0 \tag{7.4}$$

$$2x_1 - x_2 - x_3 + x_5 \geq 4 \tag{7.5}$$

$$1 \leq x \leq 3, \; x_j \in (0,1), j = 3,4,5$$

For the above MILP problem, we are interested in studying how the objective value changes with respect to the right hand side of the first constraint:

$$2x_1 + x_2 + x_3 \geq 7 + \Delta\theta$$

Following the approach presented in section 7.2.1, the following steps are applied.

Step 1: Solve the original problem with a B&B tree as shown in Figure 7.4. The optimal solution is provided by node 0 with $z = 11.5$ and $(x_3, x_4, x_5) = (0, 1, 1)$.

Step 2: Perform linear sensitivity analysis on node 0, which result in $\Delta\theta^{max} = 0$.

Stpe 3: For $\Delta\theta = \Delta\theta^{max} + \epsilon$ ($\epsilon = 0.1$), node 0 yields noninteger solution (Case 1). The updated tree is shown in Figure 7.5, where the value of dual multiplier λ is associated with each leaf node.

Step 2: Since $min\{\frac{12.1-11.8}{3-1}, \frac{12-11.8}{3-0}, \frac{12-11.8}{3-0}\} = 0.2/3$ is within the range $[0, 2]$

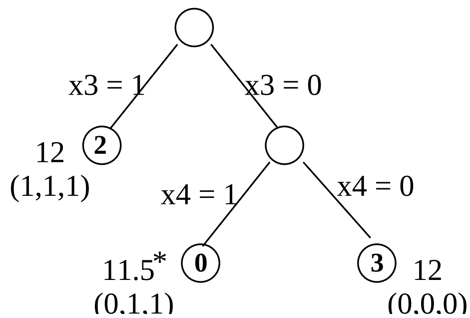


Figure 7.4: Branch and Bound Tree at Step 1 for Example 1

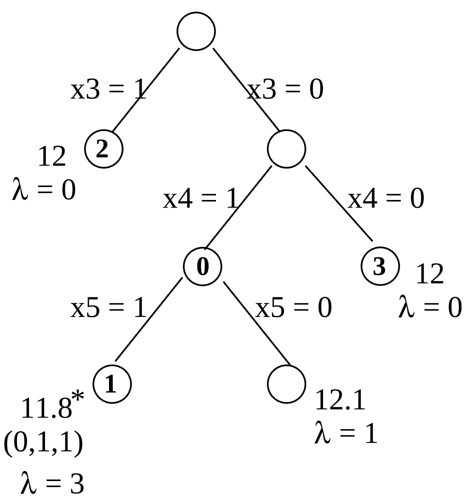


Figure 7.5: Updated Branch and Bound Tree for Example 1

obtained from sensitivity analysis, $\Delta\theta^{max}$ is equal to 0.2/3 and is given by node 2 and 3.

Step 3: For $\Delta\theta = \Delta\theta^{max} + \epsilon = 0.2/3 + 0.1/3 = 0.1$ ($\epsilon = 0.1/3$), node 1 is intersected by leaf nodes 2 and 3 (Case 2). The new optimal solution is found at the descent nodes of these two nodes, by continuing the branch and bound procedure on them, as indicated in Figure 7.6.

After two more iterations, we found that the problem becomes infeasible when $\Delta\theta$ is greater than 2. Figure 7.7 presents how the objective value and integer solution change with respect to $\Delta\theta$.

Note that the value of ϵ is usually chosen to be small enough, such as less than

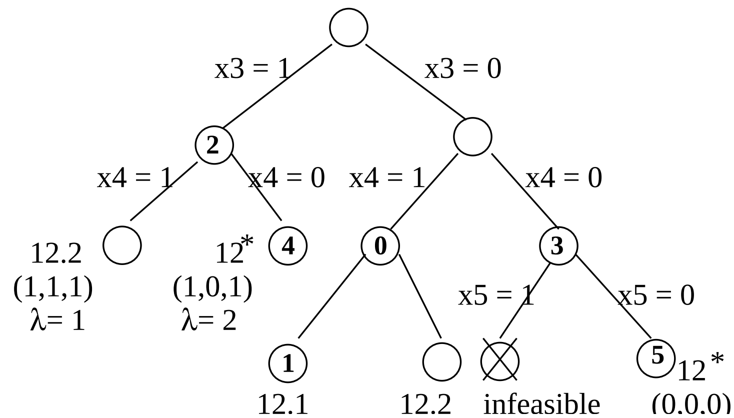


Figure 7.6: Updated Branch and Bound Tree for Example 1

2% of θ_0, so that the new $\theta_0 + \Delta\theta$ doesn't exceed the next optimal basis.

Example 2

Our second example is a scheduling problem that involves a single product production line consisting of a mixer, a reactor and a purifier as shown in Figure 7.8. The data for this example are obtained from Ierapetritou and Floudas[51]. The product demand is considered to be the uncertain parameter with the objective function of minimizing production makespan. The nominal value of demand is 50 and is expected to vary within [20, 100].

Step 1: The problem is solved at the nominal demand value (50). A branch and bound tree is constructed as illustrated in Figure 7.9, where the value next to each node indicates the objective value of relaxed LP and the binary variables that are branched at each level are shown at the right side. The optimal schedule is found to be schedule 1 with makespan 9.83 hours, and schedule 4 and 5 are feasible schedules that have makespan 10.83 hours. Note that nodes a, b, c, and d practically all represent the same schedule as node 1, which means they are equivalent schedules.

Step 2: Performing linear sensitivity analysis on node 1, we get $\Delta\theta^{max} = 0$.

Step 3: For $\Delta\theta = \Delta\theta^{max} + \epsilon$ ($\epsilon = 1$), node 1 yields noninteger solution (Case 1). The updated tree is shown in Figure 7.10, where schedule 2 provides the optimal schedule with objective value of 10.82 hours and schedule 3 is a feasible schedule.

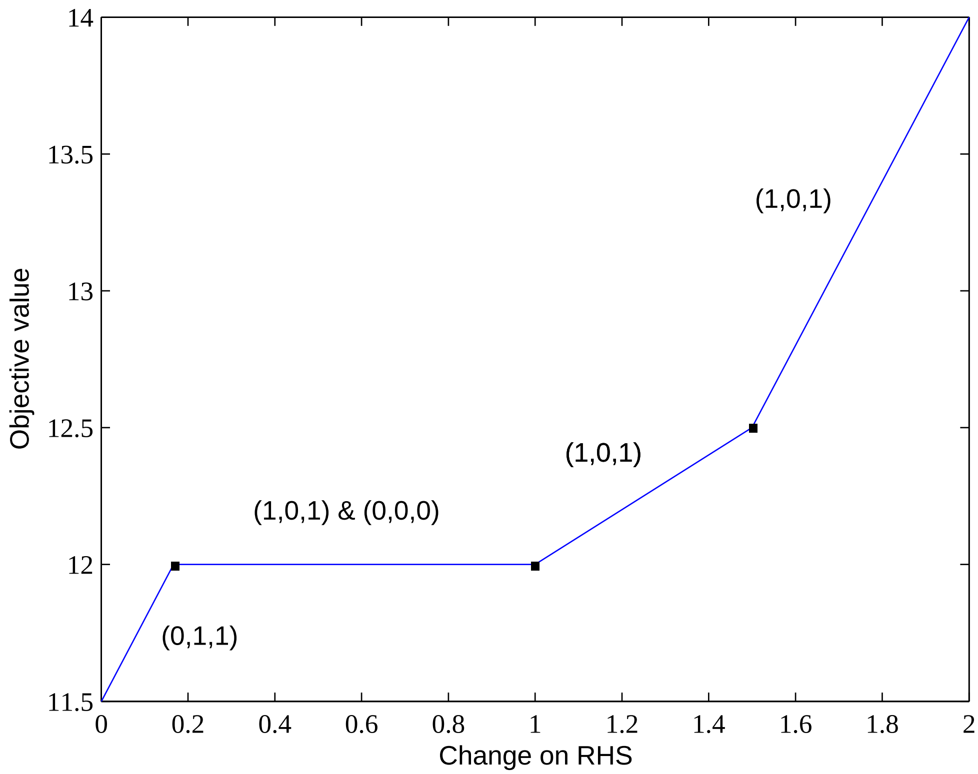


Figure 7.7: Parametric Solution for Example 1

Step 2: Sensitivity analysis on node 2 gives a range of [0, 10] for the current optimal basis. As illustrated in Figure 7.10, the dual multipliers at node 3, 4 and 5 are all greater than the one at node 2, which means node 2 will not be intersected by other nodes. Hence, $\Delta\theta^{max}$ is equal to 10.

Step 3: For $\Delta\theta = \Delta\theta^{max} + \epsilon = 10 + 1 = 11$ $(\epsilon = 1)$, node 2 still provides the optimal schedule but the basis changes (Case 3). The new optimal makespan is 11.41 hours when the demand becomes 62.

Schedule 2 continues to be the optimal schedule in the next iteration, in which $\Delta\theta^{max} = 40$. After that, the problem becomes infeasible.

After the schedules are determined to cover the demand uncertainty, the schedules are evaluated based on their robustness and expected performance in the face of uncertainty. In this work, corrected Standard Deviation robustness metric[111] is

Figure 7.8: State-Task Network Representation for Example 2

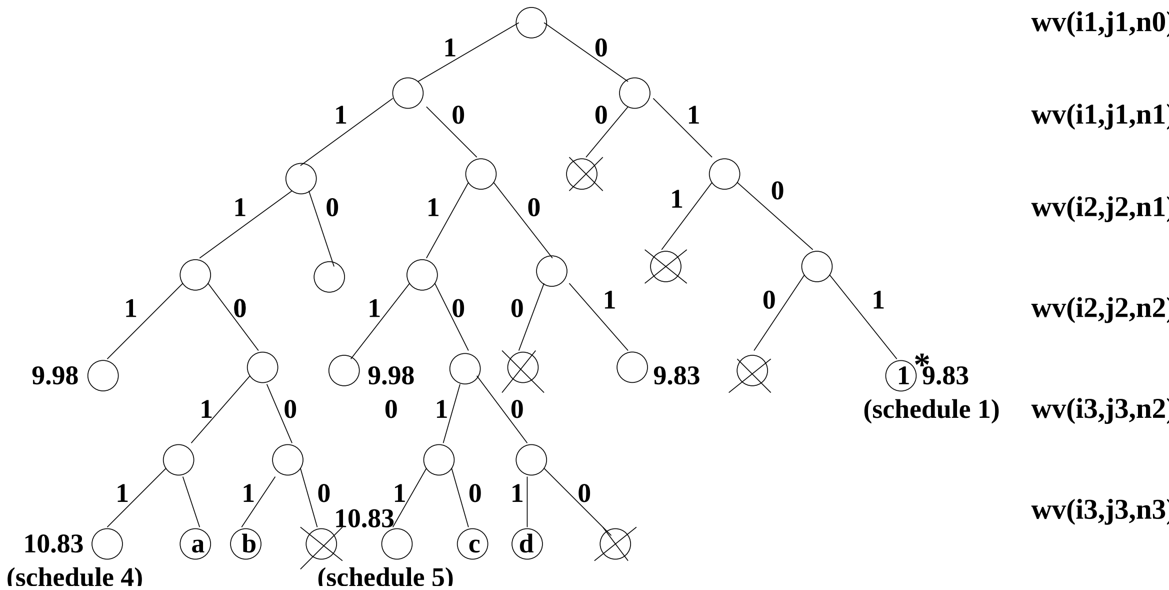


Figure 7.9: Branch and Bound Tree at Step 1 for Example 2

used considering 5 scenarios in the demand interval $[20, 100]$. The average makespan, standard deviation and the nominal makespan are shown in Table 7.1. Note that although schedule 1 provides the best solution at nominal point, it has poor performance over the entire interval. Schedule 2 has the highest robustness and provides a short average makespan. Schedule 3 provides a compromise solution with robustness between the ones of schedules 1 and 2 and average and nominal performance close to the best ones given by schedule 2. Schedule 4(5) gives the shortest average makespan but has the poorest robustness.

7.3.2 Examples for Multiple Uncertain Parameters Case

The two examples in Acevedo and Pistikopoulos's paper[1] are used for the case of multiple uncertain parameters.

	Schedule 1	Schedule 2	Schedule 3	Schedule 4 & 5
H_{nom}(h)	9.83	10.77	10.91	10.83
H_{avg} (h)	14.20	11.56	11.79	11.40
SD_{corr}	5.52	1.61	2.17	6.92

Table 7.1: Comparison of Alternative Schedules for Example 2

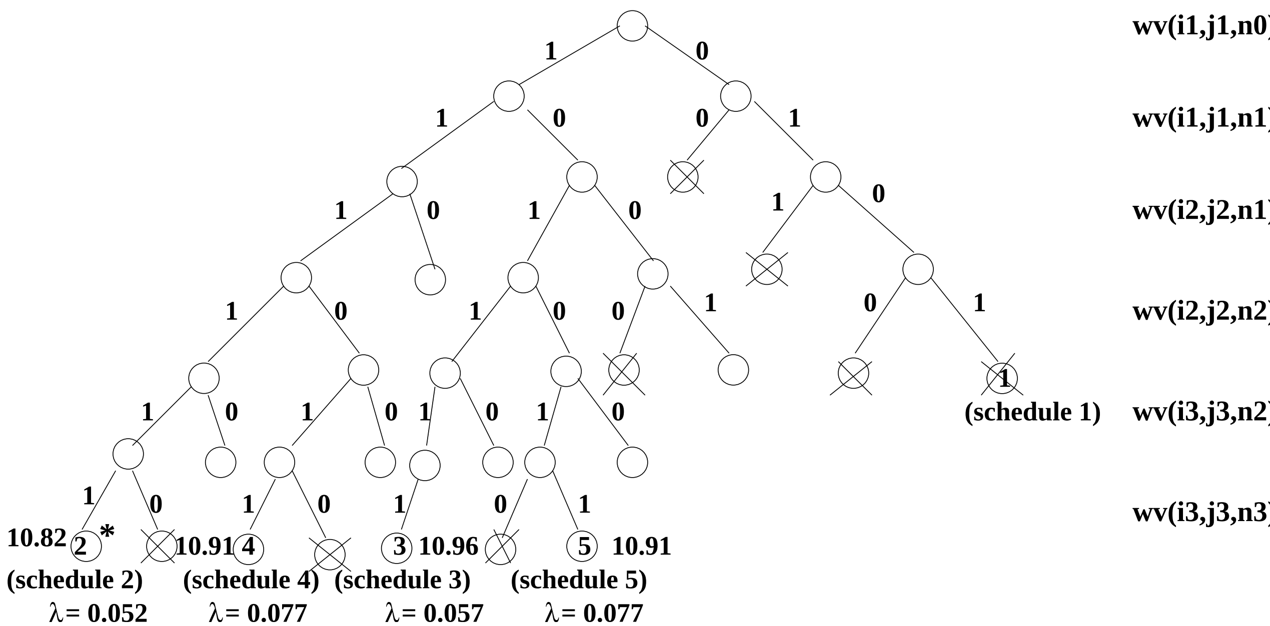


Figure 7.10: Updated Branch and Bound Tree for Example 2

Example 3

$$\min \quad z = -3x_1 - 8x_2 + 4y_1 + 2y_2$$

$$\text{subject to} \quad x_1 + x_2 \leq 13 + \theta_1 \tag{7.6}$$

$$5x_1 - 4x_2 \leq 20 \tag{7.7}$$

$$-8x_1 + 22x_2 \leq 121 + \theta_2 \tag{7.8}$$

$$4x_1 + x_2 \geq 8 \tag{7.9}$$

$$x_1 - 10y_1 \leq 0 \tag{7.10}$$

$$x_2 - 15y_3 \leq 0 \tag{7.11}$$

$$x \geq 0; y \in \{0,1\}; 0 \leq \theta_1, \theta_2 \leq 10$$

This problem contains two continuous variables x_1, x_2 and two binary variables y_1, y_2. θ_1 and θ_2 in the RHS of constraints (6) and (8) are uncertain parameters varying in the range $[0, 10]$.

According to the the approach proposed in section 7.3.2, the following steps are considered.

Step 1: Solve the original problem

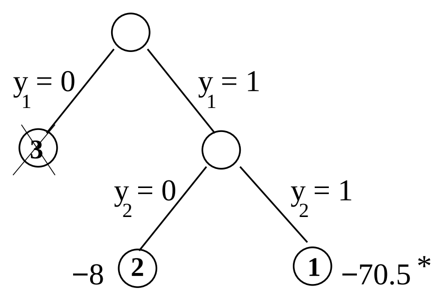


Figure 7.11: Branch and Bound Tree for Example 3

The problem is first solved with B&B algorithm for $\theta^0 = (0, 0)$. As shown in Figure 7.11, both nodes 1 and 2 provide integer solutions, which are $(y_1, y_2) = (1, 1)$ and $(y_1, y_2) = (1, 0)$, respectively, and the optimal values at these two nodes are -70.5 and -8, respectively. Therefore, node 1 gives the optimal solution. Node 3 is found to be an infeasible node and thus fanthomed.

Step 2: mpLP on the leaf nodes

Multi-parametric linear programming is performed on these two leaf nodes, so as to determine how the objective value z changes with respect to θ_1 and θ_2 and the critical regions in which the optimal functions are valid.

Node 1: The dual multipliers λ_1 and β_1 of the relaxed LP at node 1 are -4.3333 and -0.1667, thus, the optimal function at θ^0 is $z_1^{(1)} = -70.5 - 4.3333\theta_1 - 0.1667\theta_2$. In order to find the critical region for this function, problem (P4) is formulated and solved as follows:

$$\max \quad (-13-\theta_1)d_1 - 20d_2 + (-121-\theta_2)d_3 + 8d_4 - d_7 - d_8 - z$$

$$\text{subject to} \quad -d_1 - 5d_2 + 8d_3 + 4d_4 - d_5 \le -3 \qquad (7.12)$$

$$-d_1 + 4d_2 - 22d_3 + d_4 - d_6 \le -8 \qquad (7.13)$$

$$10d_5 - d_7 \le 4 \qquad (7.14)$$

$$15d_6 - d_8 \le 2 \qquad (7.15)$$

$$z \ge -70.5 - 4.3333\theta_1 - 0.1667\theta_2 \qquad (7.16)$$

$$0 \le \theta_1, \theta_2 \le 10$$

The solution of this problem identifies if there exists a point (θ_1, θ_2), at which the objective value z^* of the relaxed LP problem cannot be expressed by the current optimal functions. The solution is found to be $(\theta_1, \theta_2) = (10, 0)$ and the objective value is 16.74. The problem is solved using GAMS/BARON within 1 iteration in 0.02 CPU seconds. The same solution is obtained using GAMS/CONOPT in 6 iterations, while only a local optimal solution is found with an objective function of 0 using GAMS/MINOS. At $(\theta_1, \theta_2) = (10, 0)$, the optimal function of the LP problem at node 1 is $z_1^{(2)} = -97.0909 - 0.3636\theta_2$. Then the constraint $z \ge -97.0909 - 0.3636\theta_2$ is included in the above problem and the new objective value becomes 0. Therefore, the entire uncertain space can be covered by these two optimal functions.

The critical regions of the two optimal functions can be derived from the inequality $z_1^{(1)} \ge z_1^{(2)} \Longrightarrow 0.07333\theta_1 - 0.00333\theta_2 \le 0.45$.

Thus, we have:

$$z_1^{(1)} = -70.5 - 4.3333\theta_1 - 0.1667\theta_2$$

$$CR_1^{(1)} = \begin{cases} 0.07333\theta_1 - 0.00333\theta_2 \leq 0.45 \\ \theta_2 \leq 10 \end{cases}$$

$$z_1^{(2)} = -97.0909 - 0.3636\theta_2$$

$$CR_1^{(2)} = \begin{cases} 0.07333\theta_1 - 0.00333\theta_2 \geq 0.45 \\ \theta_1, \theta_2 \leq 10 \end{cases}$$

Node 2: The current optimal function is $z_2 = -8$, which means the objective value doesn't vary with respect to θ_1 and θ_2.

Similarly to node 1, the following problem is formulated and solved and the objective value is found to be 0.

$$\max \quad (-13-\theta_1)d_1 - 20d_2 + (-121-\theta_2)d_3 + 8d_4 - d_7 - d_8 - z$$

$$\text{subject to} \quad -d_1 - 5d_2 + 8d_3 + 4d_4 - d_5 \leq -3 \qquad (7.17)$$

$$-d_1 + 4d_2 - 22d_3 + d_4 - d_6 \leq -8 \qquad (7.18)$$

$$10d_5 - d_7 \leq 4 \qquad (7.19)$$

$$15d_6 - d_8 \leq 2 \qquad (7.20)$$

$$z \geq 8 \qquad (7.21)$$

$$0 \leq \theta_1, \theta_2 \leq 10$$

Therefore, the mpLP procedure stops and the result at node 2 is:

$$z_2 = 8 \qquad CR_2 = \begin{cases} \theta_1 \leq 10 \\ \theta_2 \leq 10 \end{cases}$$

Step 3: Compare critical regions and determine optimal functions

$\mathbf{CR_1^{(1)}}$ **and** $\mathbf{CR_2}$: Since CR_2 contains the entire uncertainty space, $CR_1^{(1)} \cap CR_2 = CR_1^{(1)}$. Then $z_1^{(1)}$ and z_2 are compared in $CR_1^{(1)}$, which can be achieved by solving the following redundancy test formulation:

$$\begin{aligned} \min \quad & \epsilon \\ -70.5 - 4.3333\theta_1 - 0.1667\theta_2 + \epsilon &= -8 \\ 0.07333\theta_1 - 0.00333\theta_2 &\leq 0.45 \\ \theta_2 &\leq 10 \end{aligned}$$

It is found that $\epsilon > 0$, therefore $z_1^{(1)} \leq z_2$.

$\mathbf{CR_1^{(2)}}$ and $\mathbf{CR_2}$: It is trivial to see that $CR_1^{(2)} \cap CR_2 = CR_1^{(2)}$. The following redundancy test shows that $\epsilon > 0$, hence constraint $z_1^{(2)} \leq z_2$ is redundant in $CR_1^{(2)}$.

$$\begin{aligned} \min \quad & \epsilon \\ -97.0909 - 0.3636\theta_2 + \epsilon &= -8 \\ 0.07333\theta_1 - 0.00333\theta_2 &\geq 0.45 \\ \theta_1, \theta_2 &\leq 10 \end{aligned}$$

Thus the final optimal solution for this problem is:

$$y* = (1, 1)$$

$$z_1(\theta) = -70.5 - 4.3333\theta_1 - 0.1667\theta_2$$

$$CR_1 = \begin{cases} 0.07333\theta_1 - 0.00333\theta_2 \leq 0.45 \\ \theta_2 \leq 10 \end{cases}$$

$$z_2(\theta) = -97.0909 - 0.3636\theta_2$$

$$CR_2 = \begin{cases} 0.07333\theta_1 - 0.00333\theta_2 \geq 0.45 \\ \theta_1, \theta_2 \leq 10 \end{cases}$$

However, the result from the literature[1] contains only z_1 and CR_1.

Example 4

The next example involves two continuous variables x_1, x_2, two binary variables y_1, y_2, and three uncertain parameters θ_1, θ_2 and θ_3 that vary within the range of

$[0, 5]$.

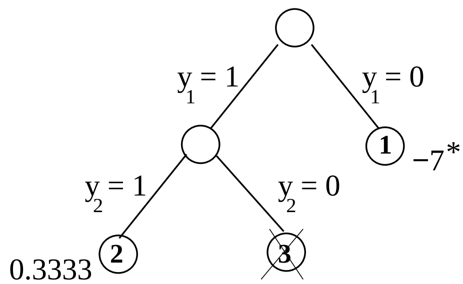


Figure 7.12: Branch and Bound Tree for Example 4

$$\min \quad z = -3x_1 - 2x_2 + 10y_1 + 5y_2$$

$$\text{subject to} \quad x_1 \leq 10 + \theta_1 + 2\theta_2 \tag{7.22}$$

$$x_2 \leq 10 - \theta_1 + \theta_2 \tag{7.23}$$

$$x_1 + x_2 \leq 20 - \theta_2 \tag{7.24}$$

$$x_1 + 2x_2 \leq 12 + \theta_1 - \theta_3 \tag{7.25}$$

$$x_1 - 20y_1 \leq 0 \tag{7.26}$$

$$x_2 - 20y_2 \leq 0 \tag{7.27}$$

$$-x_1 + x_2 \geq 4 - \theta_3 \tag{7.28}$$

$$y_1 + y_2 \geq 1 \tag{7.29}$$

$$x \geq 0; y \in \{0, 1\}; 0 \leq \theta_1, \theta_2, \theta_3 \leq 5$$

Step 1: Solve the original problem

At $\theta^0 = (0, 0, 0)$, the B&B tree results in two integer solutions $(y_1, y_2) = (0, 1)$ and $(y_1, y_2) = (1, 1)$ at node 1 and node 2, respectively. As illustrated in Figure 7.12, the optimal value is provided by node 1, which is -7. Node 3 is an infeasible node and therefore is fathomed.

Step 2: mpLP on the leaf nodes

Node 1: The optimal function at θ^0 is $z_1^{(1)} = -7 - \theta_1 + \theta_3$, which is then included in the dual of the LP problem so as to identify other optimal functions.

$$\begin{aligned}
\max \quad & (-10-\theta_1-2\theta_2)d_1+(-10+\theta_1-\theta_2)d_2+(-20+\theta_2)d_3 \\
& +(-12-\theta_1+\theta_3)d_4+(4-\theta_3)d_7+d_8-d_9-d_{10}-z \\
& \text{subject to} \quad -d_1-d_3-d_4-d_5-d_7 \le -3 \\
& -d_2-d_3-2d_4+d_7 \le -2 \\
& 20d_5+d_8-d_9 \le 10 \\
& 20d_6+d_8-d_{10} \le 5 \\
& z \ge -7-\theta_1+\theta_3 \\
& 0 \le \theta_1,\theta_2,\theta_3 \le 5
\end{aligned}$$

It is found that at $\theta = (5,0,0)$, the gap between the LP objective and the value given by the current optimal function $z_1^{(1)}$ is 7. Using GAMS/BARON as the NLP solver, the solution requires only 1 iteration and 0.02 CPU seconds to identify the global optimum, whereas a local optimal solution with objective value 0 is obtained using CONOPT and MINOS. Therefore, the optimal function at $\theta = (5,0,0)$, which is $z_1^{(2)} = -15+2\theta_1-2\theta_2$, is added to the above problem. The mpLP procedure terminates after this iteration and the results at node 1 are:

$$z_1^{(1)} = -7-\theta_1+\theta_3$$

$$CR_1^{(1)} = \begin{cases} 3\theta_1-2\theta_2-\theta_3 \le 8 \\ \theta_1,\theta_2,\theta_3 \le 5 \end{cases}$$

$$z_1^{(2)} = -15+2\theta_1-2\theta_2$$

$$CR_1^{(2)} = \begin{cases} 3\theta_1-2\theta_2-\theta_3 \ge 8 \\ \theta_1,\theta_2,\theta_3 \le 5 \end{cases}$$

Node 2: Starting from θ^0, two additional problems are solved to complete the mpLP. The optimal functions and their corresponding critical regions are as follows:

$$z_2^{(1)} = -0.3333 - 1.6667\theta_1 + 0.3333\theta_3$$

$$CR_2^{(1)} = \begin{cases} 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \leq 23.3333 \\ \theta_1, \theta_2, \theta_3 \leq 5 \end{cases}$$

$$z_2^{(2)} = -23 + 5\theta_1 - 5\theta_2 - 3\theta_3$$

$$CR_2^{(2)} = \begin{cases} 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \geq 23.3333 \\ \theta_1, \theta_2, \theta_3 \leq 5 \end{cases}$$

Step 3: Compare critical regions and determine optimal functions

$\mathbf{CR_1^{(1)}}$ **and** $\mathbf{CR_2^{(1)}}$: The intersection of $CR_1^{(1)}$ and $CR_2^{(1)}$ can be determined by combining all the constraints of these two regions:

$$\begin{cases} 3\theta_1 - 2\theta_2 - \theta_3 \leq 8 \\ 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \leq 23.3333 \\ \theta_1, \theta_2, \theta_3 \leq 5 \end{cases}$$

Then the redundancy test is performed on each of the constraints and $6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \leq 23.3333$ is found to be redundant, therefore, the intersection is:

$$CR_a^{int} = \begin{cases} 3\theta_1 - 2\theta_2 - \theta_3 \leq 8 \\ \theta_1, \theta_2, \theta_3 \leq 5 \end{cases}$$

Another redundancy test shows that the constraint $z_1^{(1)} \leq z_2^{(1)}$ is redundant in CR_a^{int}, hence, $z_1^{(1)}$ represents the optimal function in CR_a^{int}.

$\mathbf{CR_1^{(1)}}$ **and** $\mathbf{CR_2^{(2)}}$: Since $CR_1^{(1)} \cap CR_2^{(1)} = CR_1^{(1)}$, it is trivial that $CR_1^{(1)} \cap CR_2^{(2)} = \emptyset$.

$\mathbf{CR_1^{(2)}}$ **and** $\mathbf{CR_2^{(1)}}$: $CR_1^{(2)} \cap CR_2^{(1)} = CR_b^{int}$, and constraint $z_1^{(2)} \leq z_2^{(1)}$ is not redundant.

$$CR_b^{int} = \begin{cases} 3\theta_1 - 2\theta_2 - \theta_3 \geq 8 \\ 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \leq 23.3333 \\ \theta_1 \leq 5, \quad \theta_3 \leq 5 \end{cases}$$

Thus, CR_b^{int} is divided into two uncertain spaces by constraint $z_1^{(2)} \leq z_2^{(1)}$.

$\mathbf{CR_1^{(2)}}$ and $\mathbf{CR_2^{(2)}}$: $CR_1^{(2)} \cap CR_2^{(2)} = CR_c^{int}$, and constraint $z_1^{(2)} \leq z_2^{(2)}$ is not redundant.

$$CR_c^{int} = \begin{cases} 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \geq 23.3333 \\ \theta_1 \leq 5 \end{cases}$$

Similarly, CR_c^{int} is divided into two uncertain spaces by constraint $z_1^{(2)} \leq z_2^{(2)}$.

To summarize, the final results of MILP parametric analysis for this example are:

$$\text{Optimal integer solution:} \qquad y* = (0, 1)$$

$$z_1(\theta) = -7 - \theta_1 + \theta_3$$

$$CR_1 = \begin{cases} 3\theta_1 - 2\theta_2 - \theta_3 \leq 8 \\ \theta_1, \theta_2, \theta_3 \leq 5 \end{cases}$$

$$z_2 = -23 + 5\theta_1 - 5\theta_2 - 3\theta_3$$

$$CR_{2a} = \begin{cases} 3.6667\theta_1 - 2\theta_2 - 0.3333\theta_3 \leq 15.3333 \\ 3\theta_1 - 2\theta_2 - \theta_3 \geq 8 \\ 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \leq 23.3333 \\ \theta_1 \leq 5, \quad \theta_3 \leq 5 \end{cases}$$

$$CR_{2b} = \begin{cases} 3\theta_1 - 2\theta_2 - \theta_3 \geq 8 \\ 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \geq 23.3333 \\ \theta_1 \leq 5 \end{cases}$$

$$\text{Optimal integer solution:} \qquad y* = (1, 1)$$

$$z_3 = -0.3333 - 1.6667\theta_1 + 0.3333\theta_3$$

$$CR_3 = \begin{cases} 3.6667\theta_1 - 2\theta_2 - 0.3333\theta_3 \geq 15.3333 \\ 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \leq 23.3333 \\ \theta_1 \leq 5, \quad \theta_3 \leq 5 \end{cases}$$

$$z_4 = -23 + 5\theta_1 - 5\theta_2 - 3\theta_3$$

$$CR_4 = \begin{cases} 3\theta_1 - 3\theta_2 - 3\theta_3 \leq 8 \\ 6.6667\theta_1 - 5\theta_2 - 3.3333\theta_3 \geq 23.3333 \\ \theta_1 \leq 5 \end{cases}$$

The results obtained are the same as reported in the literature. Comparing the computational complexity, our proposed approach requires the solution of 2 mpLPs at two leaf nodes that solve 4 NLP problems, and 3 comparisons of the critical regions. Using the method proposed by Acevedo and Pistikopoulos[1], it requires a total of 7 mpLPs solved at 5 nodes during the B&B procedure, and 8 comparisons of the critical regions.

7.4 Summary and Future Work

The issue of uncertainty analysis for MILP problems is addressed in this chapter. An integrated framework is developed that allows the parameters in the RHS of the MILP formulation to vary independently. It mainly consists of two steps: LP/mpLP sensitivity analysis and updating the B&B tree. For the case of mpLP, a novel algorithm is proposed which solves a set of NLP problems iteratively using the commercially available global optimization solver BARON. The approach is illustrated through a number of example problems.

The proposed approach can be further developed to enable the analysis of uncertainty in the constraints coefficients and the case that uncertainty exists in the objective function cofficients, constraints coefficients and the RHS parameter at the same time.

7.4.1 Uncertain Constraint Coefficients

Assuming a perturbation of problem constraint coefficients such that:

$$A + \Delta A \geq \theta$$

A linear bilevel programming problem (BPP) is formulated as follows:

$$\begin{aligned} & \max \quad cx^* - z \\ \text{subject to} \quad & z \geq z^{(k)} + \lambda^{(k)} a_1 + \beta^{(k)} a_2, k = 1, ...K \\ & a_{10} \leq a_1 \leq a_{10} + \Delta a_1 \\ & a_{20} \leq a_2 \leq a_{20} + \Delta a_2 \\ \text{where} \quad & x^* = argmin \quad cx \\ & \text{subject to} \quad Ax \geq \theta \end{aligned}$$

where $a_1, a_2 \in A$ are two uncertain parameters.

Bilevel programming has been proved to be NP-hard problem. A number of algorithms have been developed so far for solving bilevel programming problem. Detailed reviews on the alogirthms developed for bilevel programming problems can be found in Bard's book[10] and Vicente and Calamai[110]. The most popular method for solving the linear BPP is known as "Kuhn-Tucker" approach[37,11,12,47]. For a general linear BPP has the form:

$$\begin{aligned} \textbf{(P6)} \qquad & \min_{x \in X} \quad F(x,y) = c_1 x + d_1 y \\ \text{subject to} \quad & A_1 x + B_1 y \leq b_1 \\ & \min_{y \in Y} \quad f(x,y) = c_2 x + d_2 y \\ \text{subject to} \quad & A_2 x + B_2 y \leq b_2 \end{aligned}$$

The fundamental strategy of Kuhn-Tucker approach is to replace the lower-level problem with its KKT condition and add it to the upper-level pr oblem. A necessary condition that (x^*, y^*) solves problem (P6) is that there exist vector u^*, v^* such that (x^*, y^*, u^*, v^*) solves:

$$
\begin{aligned}
\textbf{(P7)} \qquad & \min\ c_1x + d_1y \\
\text{subject to} \quad & A_1x + B_1y \leq b_1 \\
& uB_2 - v = -d_2 \\
& u(b_2 - A_2x - B_2y) + vy = 0 \qquad (7.30) \\
& A_2x + B_2y \leq b_2 \\
& x \geq 0, y \geq 0, u \geq 0, v \geq 0
\end{aligned}
$$

The basic idea is to use a branch and bound strategy to deal with the complementarity constraint (7.30), since this problem is linear except for the complementary slackness term in (7.30). Bard and Falk[11] proposed an approach that rewrites the the nonlinear term as the sum of piecewise linear, separable functions and then uses a golbally convergent nonlinear code to find solutions. Later, Bard and Moore developed an implicit approach that omits the constraint (7.30) and solve the resulting linear subproblem. At each iteration, it checks if constraint (7.30) is satisfied. If so, the corresponding point is a potential solution to (P6). If not, a B&B scheme is used to examine all combinations of complementary slackness. This method is approved to be much more efficient.

7.4.2 Uncertainties in c, A, and θ

When uncertainties exist in the ojective function coefficients c, constraint coefficients A, and RHS parameters θ at the same time, following bilevel programming problem is formulated:

$$\max \quad (c+\Delta c)x^* - z$$

$$\text{subject} \quad \text{to} \quad z \geq z^{(k)} + \alpha^{(k)}a_1 + \beta^{(k)}a_2 + \gamma^{(k)}c_1$$

$$+\lambda^{(k)}c_2 + \mu^{(k)}\theta_1 + \nu^{(k)}\theta_2, k = 1, ...K$$

$$a_{10} \leq a_1 \leq a_{10} + \Delta a_1$$

$$a_{20} \leq a_2 \leq a_{20} + \Delta a_2$$

$$c_{10} \leq c_1 \leq c_{10} + \Delta c_1$$

$$c_{20} \leq c_1 \leq c_{20} + \Delta c_2$$

$$\theta_{10} \leq \theta_2 \leq \theta_{10} + \Delta \theta_1$$

$$\theta_{20} \leq \theta_2 \leq \theta_{20} + \Delta \theta_2$$

$$\text{where} \quad x^* = argmin \quad (c+\Delta c)x$$

$$\text{subject} \ \ \text{to} \ \ (A+\Delta A)x \geq \theta + \Delta\theta$$

The objective functions in both upper and lower level prolems are nonlinear. The KKT optimality conditions become only necessary when nonconvexities are involved in the lower level problem. Global optimality is guaranteed for general nonlinear bilevel problems that involve twice differentiable functions. In general, methods for solving nonlinear bilevel programming problems can be divided into three categories: branch and bound method[33,4], descent method[99,62], and penalty method[3,54]. Edmunds and Bard extended the results of Bard and Moore[12] on linear BPP to the case where the lower-level objective is quadratic, The basic idea is to relax the complementary slackness conditions and solves the res ulting problem. If the solution violates one or more of the constraints, a hybrid branch and bound algorithm is used to enumerate all possibil ities.

Gumus and Floudas[45] developed a global optimization approach which is based on the relaxation of the feasible region by convex underestimation, embedded in a branch and bound framework utilizing αBB algorithm. Global optimality is guaranteed for general nonlinear bilevel problems that involve twice differentiable functions.

Assuming that for any upper-level variables the optimal solution of the lower-level problem is unique and lower-level variables can be defined as an implicit function of the upper-level variables, the bilevel problem can be viewed solely in terms o f the upper-level variables. Given a feasible point, descent methods attempt to find a direction along which the upper-level objective decreases. Savard and Gauvin[99] developed a descent algorithm for the nonlinear bilevel programming problems. At each feasible point, the steepest descent direction is obtained by solving a quadratic bilevel programming problem. Ishizuka and Aiyoshi[53] proposed a double penalty method in which they use the augmented lower-level objective and the penalty function of Aiyoshi and Shimizu[3] but replace the lower-level problem by its stationarity condition, thus transforming the bilevel problem into a single-level program. For a given value of the penalty parameter, this single-level problem is solved using a second penalty function applied to the upper-level objective. Recently, Colson et al.[?] proposed a trust-region method whose underlying idea is: at each iteration, a model is built around the incumbent solution and minimized within some prescribed region where the approximation is thought to be a sufficiently accurate representation of the original problem. This yields a new point and if the objective function is improved, then the model is good and the trust region may be enlarged, otherwise the region is shrinked and a new model is computed.

Chapter 8

Refinery Scheduling Considering Demand Variability

8.1 Introduction

As discussed in chapters 2 and 4, substantial amount of work has been done in the area of refinery operations. The availability of LP-based commercial software for refinery production planning, such as RPMS (Refinery and Petrochemical Modeling System)[19], and PIMS (Process Industry Modeling System)[14] has allowed the development of general production plans of the whole refinery. On the other hand, very few optimization based formulations are presented for the refinery scheduling due to the lack of rigorous models to handle specific plant characteristics.

As an attempt towards integrating all the stages in refinery operations, in chapters 2 - 4, we developed comprehensive mathematical programming models for the efficient scheduling of oil-refinery operations. The overall problem is spatially decomposed into three domains: the crude-oil unloading and blending, the production-unit operations and the product blending and delivery, as shown schematically in Figure 1. Each of those sub-problems is modeled and solved in an efficient way using continuous time representation that reduces the overall number of variables and constraints.

In most of the existing refinery scheduling studies, the problem data are assumed to be deterministic. However, uncertainty arises in realistic refinery parameters such as production recipes, processing times, and order requirements due to lack of accurate process models and variability of process and environment data. The variations of certain parameters in refinery, such as feedstock qualities and yield level are of great importance in operation decisions. In oil industry, the prices and demands of crude oil, gasoline and diesel also fluctuate greatly. In order to provide a methodology that accurately models operating conditions and yields a robust solution, it is necessary to evaluate and to accommodate the realities of uncertainty. However, the issue of uncertainty in refinery operations has not been sufficiently studied for scheduling and planning problems due to the high complexity of the deterministic case. Li et al.[67] addresses the problem of refinery planning under uncertainty. A general formulation for revenue and cost calculations is proposed by considering uncertainty in raw material availability and product demand. Different standard loss function approximation methods are compared and integrated into the planning model. Neiro and Pinto[76] developed a model based on a nonlinear programming formulation. The model first incorporates multiple planning periods and the selection of different crude oil types. Uncertainty related to crude oil and product prices as well as demand is then included as a set of discrete events with specific probabilities. The aim of this chapter is to (a) help the decision maker to select the optimal overall solution; and (b) to explore the effects of uncertainty on the objective function which consists of operation cost, inventory or overall production, as well as the best way to react to uncertain parameter variability.

In this chapter, we propose to address the problem of uncertainty in refinery scheduling. Depending on the information that you have regarding uncertain parameter variability, the multiobjective robust optimization as presented in chapter 5 can be applied if there are sufficient data to select and quantify specific scenarios, otherwise the parametric MILP approach described in chapter 7 is utilized to determine the ranges of uncertain parameters where the schedules remain feasible and optimal.

The mathematical models of crude oil unloading and mixing, gasoling blending and distribution problems in chapter 2 and 4 are adopted here. In the next section, three refinery scheduling problems are presented to illustrate the effectiveness of the

Order	Amount
o1	8 ± 50%
o2	3
o3	3
o4	8 ± 50%
o5	3
o6	8 ± 50%
o7	3
o8	88 ± 50%
o9	3
o10	100 ± 50%

Table 8.1: Amount of Orders for Example 1

proposed approaches. The first two examples use NBI method to obtain Pareto optimal solutions, whereas for the third example, the MILP uncertainty analysis approach is applied to study the effects of uncertain parameter on the optimal solution behavior. Section 8.3 summarizes the work and presents some of the ideas for future developments.

8.2 Case Studies

Example 1: The first example addresses the operation schedule of gasoline blending and distribution. It involves 5 products being produced and stored in 11 storage tanks, and then lifted to satisfy 10 orders in a time horizon of 8 days. The demands of 5 of the orders are considered to be uncertain parameters with a variability of $\pm 50\%$, as illustrated in Table 8.1. Two demand scenarios of each order are considered, leading to a total of 32 scenarios. The problem is modeled using the formulation presented in section 2.2 incorporating the 32 scenarios and the following constraint is included in order to measure the expected positive deviation from the average inventory and production (solution robustness).

$$obj2 = \sum_k p^k \delta^k, \quad \delta^k = max(0, obj1^k - \sum_k p^k obj1^k)$$

where p^k represents the probability of scenario k and δ^k corresponds to the positive deviation of inventory and production ($obj1^k$) under scenario (k) from the expected value. Thus, a two-objective formulation is developed to minimize the expected total inventory and production(objective 1), and the solution robustness (objective 2).

The two anchor points obtained by solving two single objective function problems individually are:

$$f_1(x^*) = \begin{pmatrix} 1112.76^* \\ 30.27 \end{pmatrix} \quad \text{and} \quad f_2(x^*) = \begin{pmatrix} 1234.36 \\ 0^* \end{pmatrix}$$

Consequently, the utopia point is: $F^* = \begin{pmatrix} 1112.76 \\ 0 \end{pmatrix}$ and the matrix $\Phi = \begin{pmatrix} 0 & 121.6 \\ 30.27 & 0 \end{pmatrix}$
Choosing a step size of $\delta_1 = 0.05$, ω_1 and ω_2 take the following values:

$$\omega_1 = [0, 0.05, 0.1, 0.15, 0.2, ..., 1]$$
$$\omega_2 = [1, 0.95, 0.9, 0.85, ..., 1 - \omega_1]$$

The Pareto optimal surface is illustrated in Figure 8.1, where a number of evenly distributed Pareto optimal points are obtained through NBI technique. Thus, in the face of order uncertainty, a number of solutions can be achieved that correspond to different values of average total inventory and production and positive deviation. For example, at point A ($(\omega_1, \omega_2) = (0.25, 0.75)$) in Figure 8.1, the average total inventory and production in the face of uncertainty is 1192.4 and the average positive deviation from the mean is 4.68. At point B ($(\omega_1, \omega_2) = (0.8, 0.2)$), the two objective values are found to be 1129.9 and 22.41. The distribution schedules at these two points show that large orders, such as order 8, is processed three times and order 10 is processed twice, while the rest are processed once only. However, the tank utilization of schedule A is splited among tanks 3, 4, 5, 7, 10, 11, while schedule

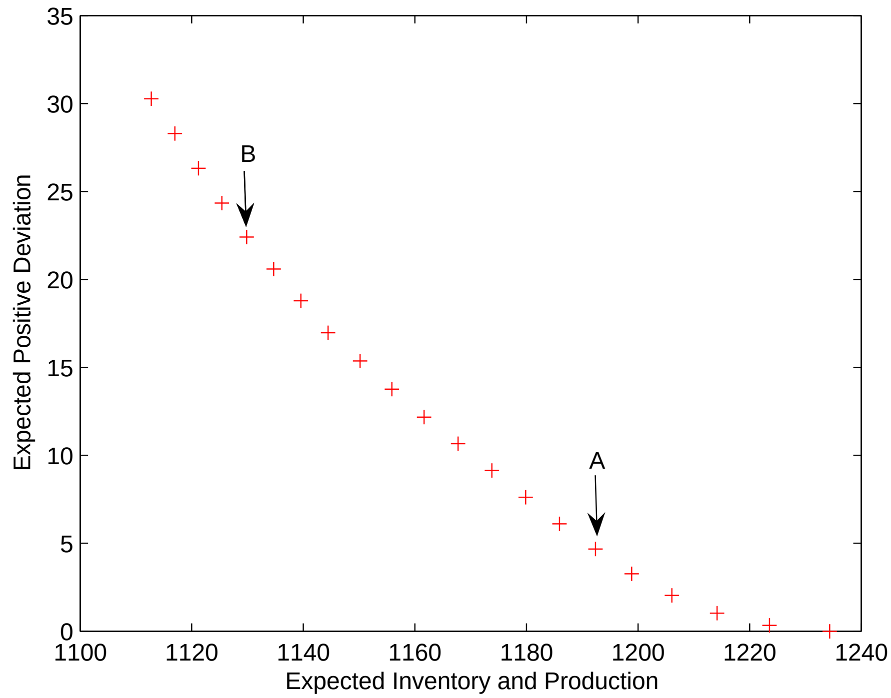


Figure 8.1: Pareto Optimal Surface for Example 1

B shows that the orders mostly take place in tanks 6 and 10. These two operation schedules at the nominal demands are presented in Figure 8.2 and 8.3, respectively. To satisfy the same amount of demand, the total inventory and production is 1161.3 using schedule A, while schedule B only needs 1053.8. However, schedule A is more flexible to accommodate order fluctuations without large deviations at the inventory and production. Consequently, a risk averse decision maker would consider schedule A as a better choice.

Example 2: The multiobjective robust optimization approach is also applied to a crude oil unloading and mixing problem as discussed in section 2.1. The data and crude oil flow network are as illustrated in Table 8.2 and Figure 8.4, respectively. Assuming $\pm 50\%$ variability of demand for crude-oil mix in both charging tanks, 5 demand scenarios $(30, 45, 60, 75, 90)$ are used to describe the uncertainty of each tank, leading to a total of 25 scenarios. Similarly, a two-objective formulation is developed to minimize the expected total operation cost (objective 1), which includes sea waiting cost, unloading cost, inventory cost of storage tanks and charging tanks, and the expected positive deviation (objective 2) from the average cost.

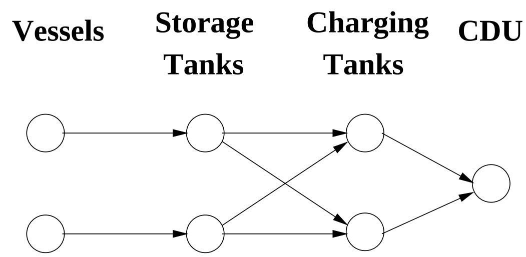


Figure 8.4: Oil Flow Network for Example 2

The resulted Pareto optimal surface is shown in Figure 8.5. As we expected, a number of evenly distributed Pareto optimal points are obtained. Moving from any point to another, the average cost cannot be reduced without decreasing the expected positive deviation, and vice versa. For example, at point A $((\omega_1, \omega_2) = (0.05, 0.95))$, the unloading schedule obtained at the nominal values of uncertain demands shows that vessel 1 should start unloading from day 1 and finish on day 3, while vessel 2 should unload from day 5 to day 7. Storage tank 1 is scheduled to transfer crude oil mix to charging tank 1 on day 7 and day 9, while storage tank 2 delivers oil mix to charging tank 1 on day 7, and to charging tank 2 on day 6 and 9. The CDU is charged by charging tank 1 on day 1, 8, 10 and by charging tank 2 on day 1, 7, 10. At this point, the average cost in the face of uncertainty is 87.232 and the average metric for robustness is 0.055. At point B $((\omega_1, \omega_2) = (0.8, 0.2))$, these two objective values are found to be 66.19 and 2.414. The nominal unloading schedules of the two vessels are the same as point A but the schedules of the other transfer are different. In particular, storage tank 1 transfers crude oil mix to charging tank 1 on day 2 and day 4, and storage tank 2 transfers oil mix to charging tank 1 on day 2, and to charging tank 2 on day 5 and 7. The CDU is charged by charging tank 1 on day 1, 3, 5 and by charging tank 2 on day 1, 6, 8. It should be noticed that although schedule B has a smaller expected cost which means that can fulfill the same demand scenarios

Scheduling Horizon(# of unit time)			10
Number of Vessel Arrivals			2
	Arrival Time	Amount of Crude	Concentration of key components
Vessel 1	1	60	0.01, 0.04
Vessel 2	5	60	0.03, 0.02
Number of Storage Tanks			2
Storage Tanks	Capacity	Initial Oil Amount	Concentration of key components
Tank 1	100	20	0.01, 0.04
Tank 2	100	10	0.03, 0.02
Number of Charging Tanks			2
Charging Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min, max)
Tank 1	100	20	0.0167 (0.01, 0.02) 0.0333 (0.03, 0.038)
Tank 2	100	20	0.03 (0.025, 0.035) 0.023 (0.018, 0.027)
Number of CDU			1
Unit costs involved in vessel operation			Unloading cost: 8, Sea waiting cost: 5
Tank inventory unit costs			Storage tank: 0.05, Charging tank: 0.08
Demand of mixed oils by CDUs			Oil mix 1: 60, Oil mix 2: 60

Table 8.2: System Information for Example 2

with lower cost comparing to schedule A, it exhibits larger variance indicating higher variability of schedules with respect to different scenarios.

Example 3: The third example is a crude oil unloading and mixing problem, which considers 1 crude-oil vessel, 1 storage tank, 2 charging tanks and 1 CDU. The data and crude oil flow network are as illustrated in Table 8.3 and Figure 8.6, respectively. The demand from charging tank 2 is considered to be the uncertain parameter, which is expected to vary within [60, 100]. The mathematical formulation presented in section 2.1 is used here.

The basic methodology presented in section 3.2 is applied in this example as follows.

Step 1: The problem is solved at an initial demand value (60) with branch and bound solution method. The objective value and the dual multiplier at the leaf nodes of the B&B tree are collected. The optimal schedule is found to be schedule A with cost 35.87 as shown in Table 8.4.

Step 2: Performing linear sensitivity analysis on node A, we get $\Delta\theta^{max} = 0$.

Step 3: For $\Delta\theta = \Delta\theta^{max} + \epsilon = 1$ ($\epsilon = 1$), the tree is updated and it is found that

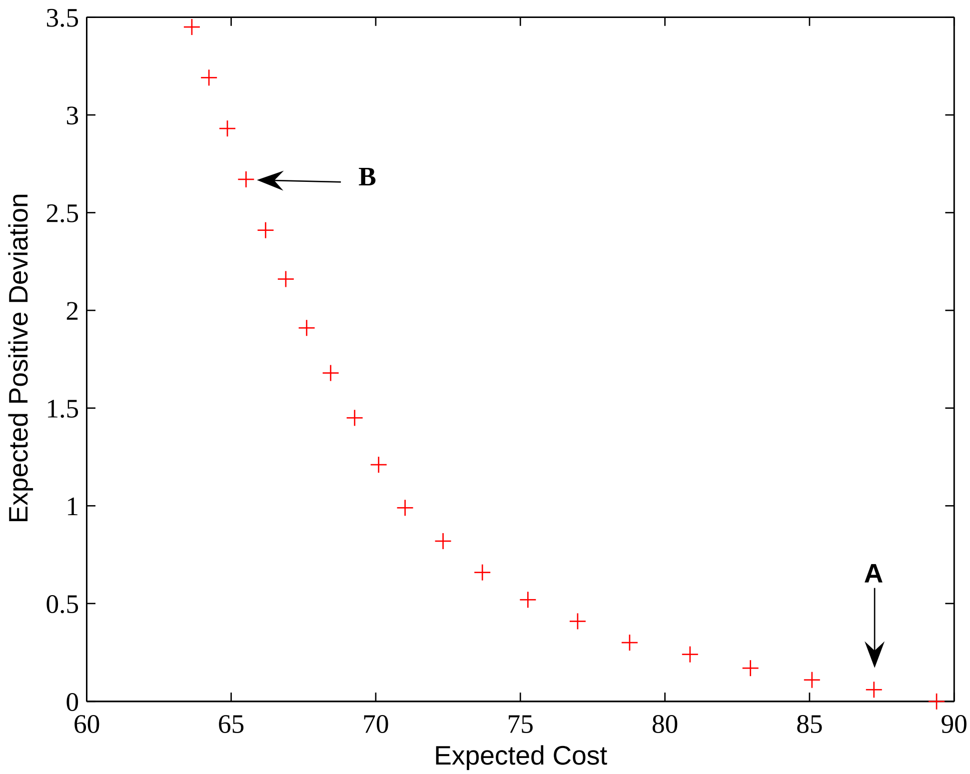


Figure 8.5: Pareto Optimal Surface for Example 2

schedule B provides the optimal schedule with objective value of 36.74.

Step 2: Sensitivity analysis on node B gives the current optimal basis $\Delta\theta^{basis} = 39$. the value of $\dfrac{z^p - z^0}{\lambda^0 - \lambda^p}$ is calculated at other 16 leaf nodes and the minimum value of those nodes is 16.5, which is less than 39. That means node B could be intersected by the node that provides the minima. Hence, $\Delta\theta^{max}$ is equal to 16.5.

Step 3: For $\Delta\theta = \Delta\theta^{max} + \epsilon = 17$ ($\epsilon = 0.5$), node B still provides the optimal schedule and the new optimal cost is 36.91 when the demand of charging tank 2 becomes 78.

Schedule B continues to be the optimal schedule in the next iteration, in which $\Delta\theta^{max} = 22$ (the demand is equal to 100). After that, the problem becomes infeasible. Therefore, schedule A provides the optimal solution when the demand from charging tank 2 is equal to 60. Schedule B becomes the optimal for the range $[60, 100]$, and

Scheduling Horizon(# of unit time)			8
Numberof Vessel Arrivals			1
	Arrival Time	Amount of Crude	Concentration of key components
Vessel 1	1	80	0.02
Number of Storage Tanks			1
Storage Tanks	Capacity	Initial Oil Amount	Concentration of key components
Tank 1	100	20	0.02
Number of Charging Tanks			2
Charging Tanks	Capacity	Initial Oil Amount	Initial comp. concentration (min, max)
Tank 1	100	20	0.02 (0.015, 0.025)
Tank 2	100	40	0.04 (0.015, 0.045)
Number of CDU			1
Unit costs involved in vessel operation			Unloading cost: 8, Sea waiting cost: 5
Tank inventory unit costs			Storage tank: 0.05, Charging tank: 0.08
Demand of mixed oils by CDUs			Oil mix 1: 60, Oil mix 2: 60

Table 8.3: System Information for Example 3

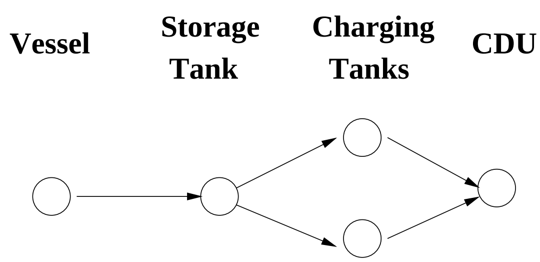


Figure 8.6: Oil Flow Network for Example 3

the linear function of how objective (inventory and production) value changes with respect to the demand is: $z = 36.73 + 0.01\Delta\theta$. Schedules A and B at the nominal demand $(60, 60)$ are shown in Figure 8.7 and 8.8. It should be pointed out that in schedule A, the oil mix in storage tank is transferred to charging tank 2 first and then to charging tank 1, and the charging sequence to CDU is tank 1, tank 2, and finally tank 1. However, these sequences are exactly the opposite in schedule B.

8.3 Summary and Future Work

This chapter addresses the problem of refinery scheduling under uncertainty through two approaches: multiobjective robust optimization and parametric mixed integer linear programming (pMILP). The robust optimization model considers a number of uncertainty scenarios, and normal boundary intersection (NBI) method is utilized to determine the Pareto optimal surface in the objective space, on which each point represents a trade-off between the original objective (minimal cost or inventory) and the solution robustness. The parametric MILP approach investigates how the value of cost or inventory and optimal schedule change with respect to the uncertain parameter. Three case studies considering the optimal operations of crude oil unloading and mixing, and gasoline blending and distribution are presented to illustrate the importance of considering uncertainty in demand in refinery scheduling operations and the results of the two proposed approaches.

In order to improve the efficiency of the multiobjective framework one direction would to be to develop an efficient scenario selection process, so that only the critical scenarios are considered and the computational complexity is not increased substantially compared with the deterministic problem. In addition, the proposed parametric MILP approach can be further developed one direction would to be to develop an efficient scenario selection process,by improving the retrieval and storage of the required dual information of the leaf nodes in the branch and bound tree. This will enable the utilization of the approach in large scale realistic problems.

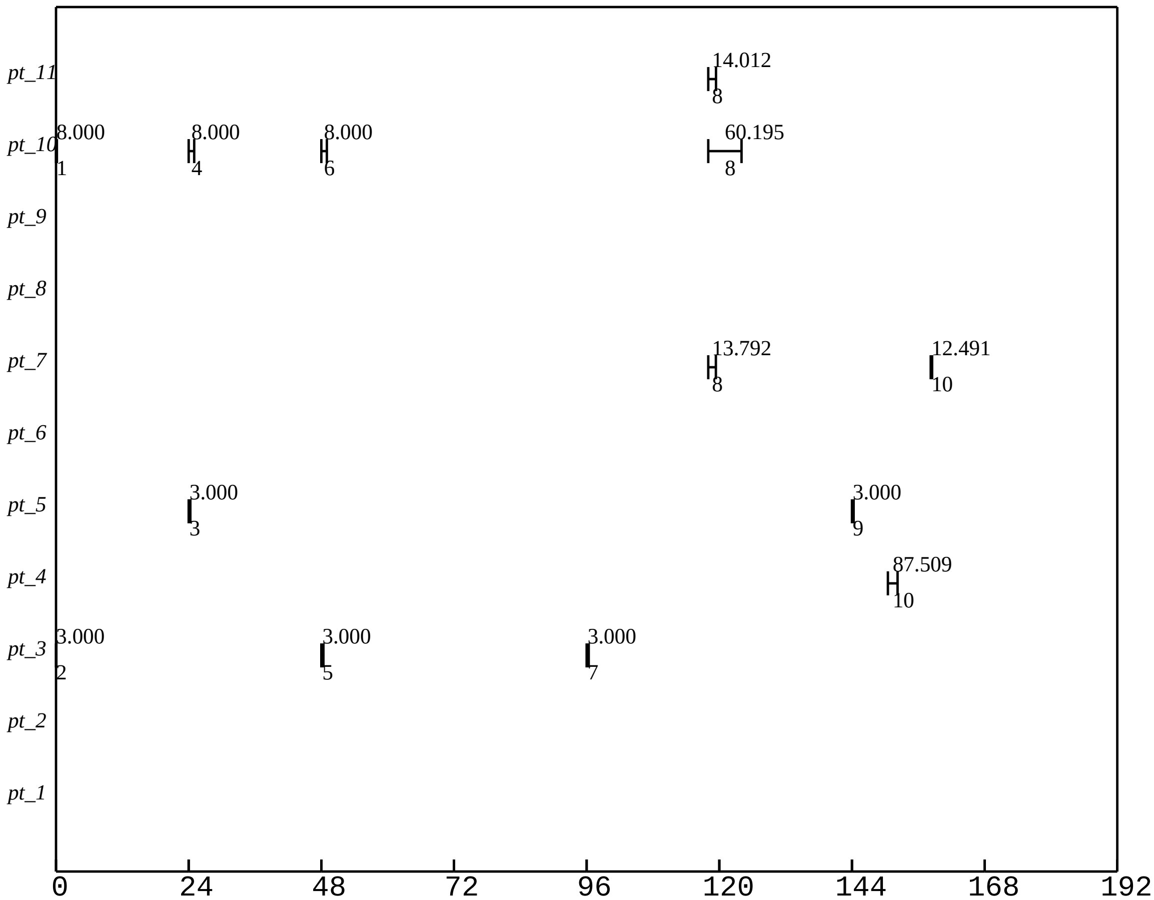


Figure 8.2: Gantt Chart of Operation Schedule A at Nominal Demand

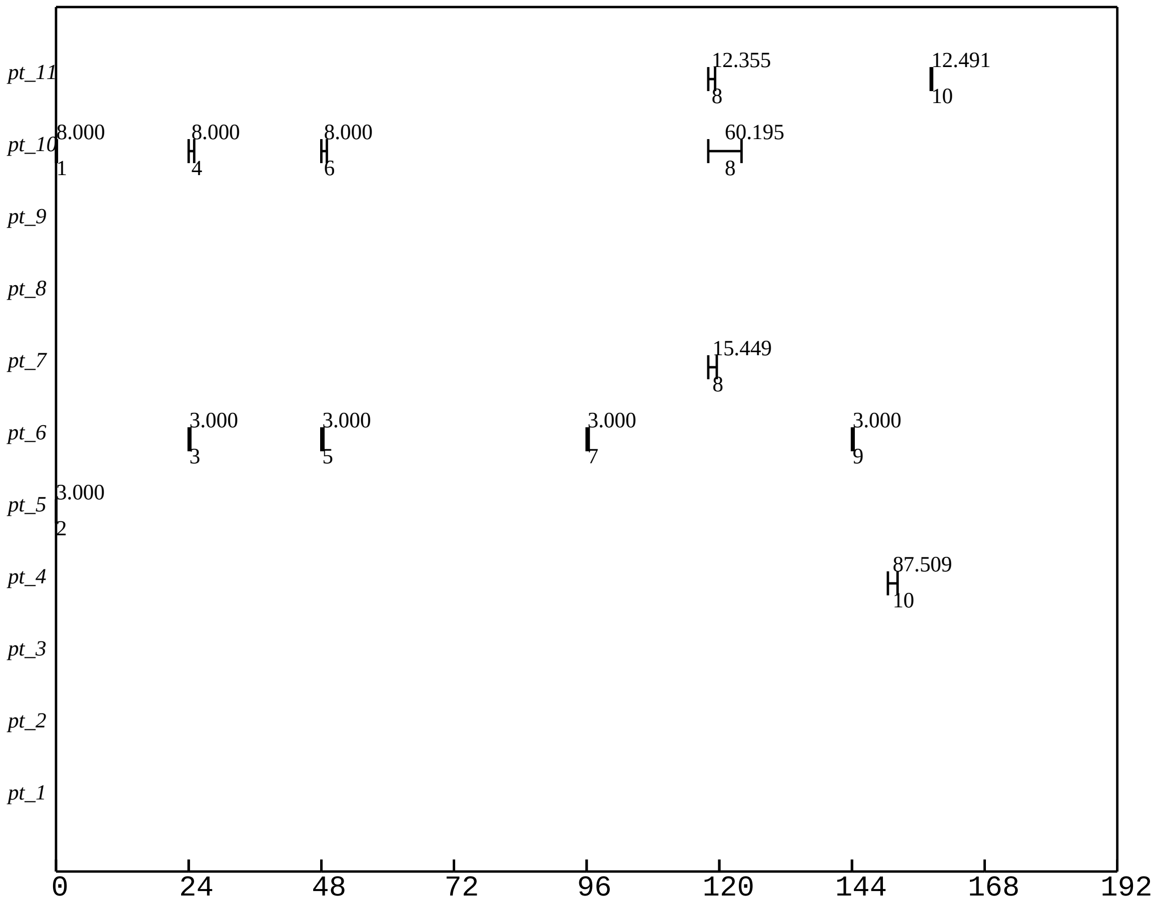


Figure 8.3: Gantt Chart of Operation Schedule B at Nominal Demand

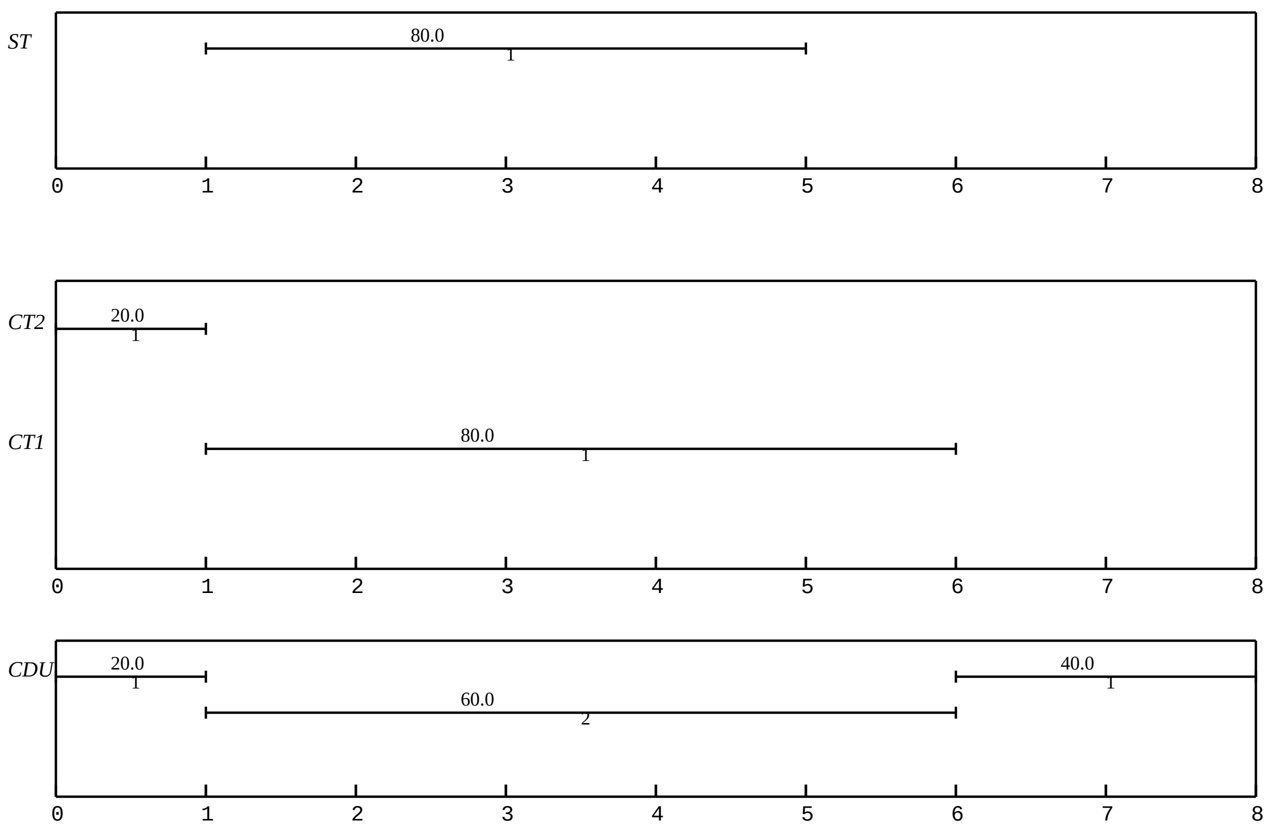


Figure 8.7: Gantt Chart of Operation Schedule A at Nominal Demand

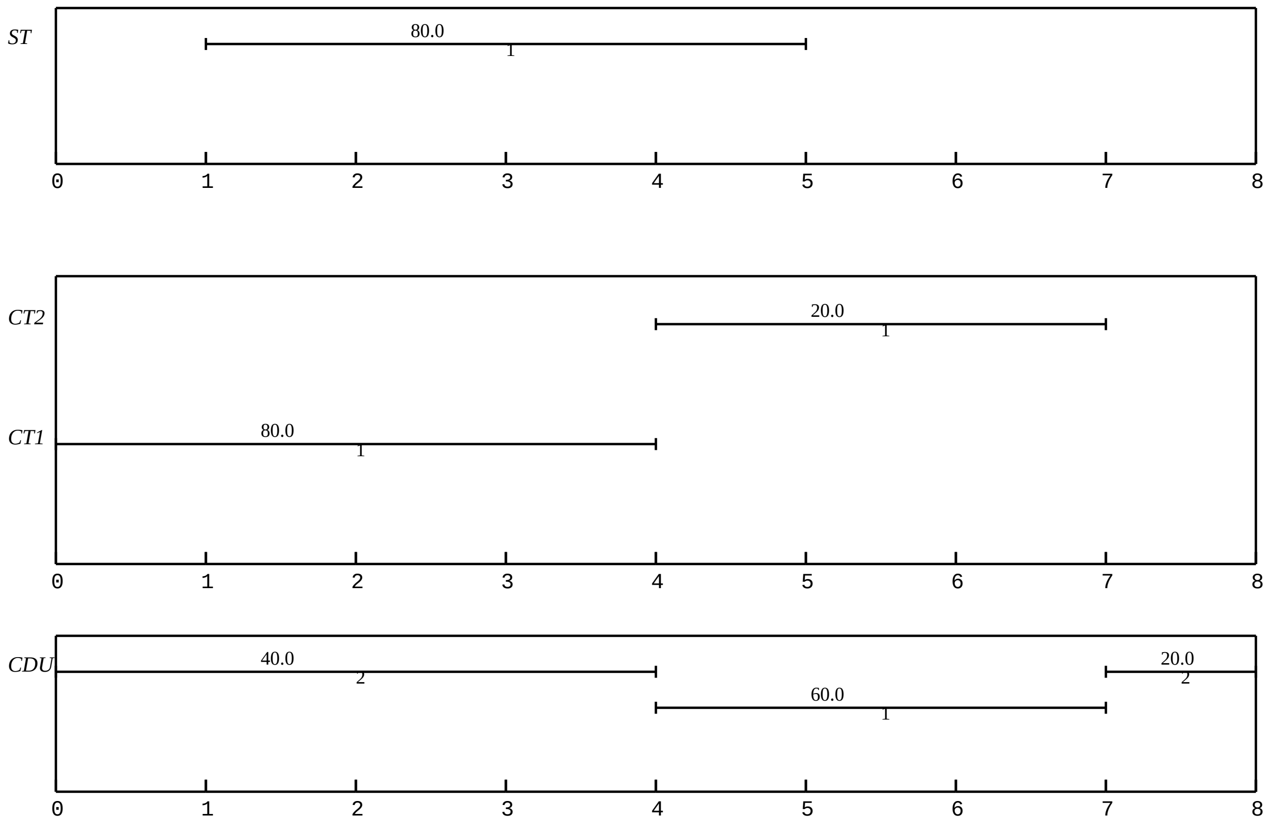


Figure 8.8: Gantt Chart of Operation Schedule B at Nominal Demand

(Task, Unit)	n0	n1	n2
(Vessel, Storage Tank)	0	1	0
(Storage Tank, Charging Tank 1)	0	1	0
(Storage Tank, Charging Tank 2)	1	0	0
(Charging Tank 1, CDU)	1	0	1
(Charging Tank 2, CDU)	0	1	0

(Schedule A)

(Task, Unit)	n0	n1	n2
(Vessel, Storage Tank)	1	0	0
(Storage Tank, Charging Tank 1)	1	0	0
(Storage Tank, Charging Tank 2)	0	1	0
(Charging Tank 1, CDU)	0	1	0
(Charging Tank 2, CDU)	1	0	1

(Schedule B)

Table 8.4: Values of Binary Variables of Optimal Schedules for Example 3

Bibliography

1. Acevedo, J., Pistikopoulos, E.N. A multiparametric programming approach for linear process engineering problems under uncertainty. *Ind. Eng. Chem. Res.* 1997, 36, 717.

2. Ahmed, S.,Sahinidis, N. Robust process planning under uncertainty. *Ind. Eng. Chem. Res.* 1998, 37, 1883.

3. Aiyoshi, E., Shimizu, K. A solution method for the static constrained Stackelberg problem via penalty method. *IEEE Transactions on Automatic Control* 1984, 29, 1111.

4. F.A. Al-Khayal, R. Horst, P. M. Pardalos. Global optimization of concave functions subject to quadratic constraints: An application in nonlinear bilevel programming. *Annals of Operations Research* 1992, 34, 125.

5. Alves, M.J., Climaco, J. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. *Eur. J. Oper. Res.* 2000, 124, 478.

6. Balasubramanian, J.,Grossmann, I.E. A novel branch and bound algorithm for scheduling flowshop plants with uncertain processing times. *Comput. Chem. Eng.* 2002, 26, 41.

7. Balasubramanian, J.,Grossmann, I.E. Scheduling optimization under uncertainty - an alternative approach. *Comput. Chem. Eng.* 2003, 27, 469.

8. Balasubramanian, J.,Grossmann, I.E. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. *Ind. Eng. Chem. Res.* 2004, 43, 3695.

9. Bansal, V., Perkins, J.D., Pistikopoulos, E.N. Flexibility analysis and design using a parametric programming framework. *AICHE J.* 2002, 48, 2851.

10. Bard, J.F. Practical bilevel optimization: algorithms and applications. *Kluwer Academic Publishers* 1998.

11. Bard, J.F., Falk, J.E. An explicit soluton to the multi-level programming problem. *Computers and Operations Research* 1982, 9, 77.

12. Bard, J.F., Moore, J.T. A branch and bound algorithm for the bilevel programming problem. *SIAM Journal of Scientific and Statistical Computing* 1990, 11, 281.

13. Basset, M.H., Perkny, J.F., Reklaitis, G.V. Obtaining realistic production plans for a batch production facility. *Comput. Chem. Eng.* 1997, 21, S1203.

14. Bechtel. PIMS (Process Industry Modeling System) User's manual. *Houston, TX: Bechtel Corp.* 1993.

15. Bertsimas, D., Georgia, P., Tayur, S. A new algebraic geometry algorithm for integer programming. *Manage. Sci.* 2000, 46, 999.

16. Bodington, C.E. Inventory management in blending optimization: Use of nonlinear optimization for gasoline blend planning and scheduling. *ORSA/TIMS National Meeting, San Francisco* 1992.

17. Bodington, C.E. Planning, Scheduling, and control integration in the process industry. *McGraw Hill, New York* 1995.

18. Bonfill, A., Espuna, A., Puigjaner, L. Addressing robustness in scheudling batch processes with uncertain operations times. *Ind. Eng. Chem. Res.* 2005, 44, 1524.

19. Bonner, Moore. RPMS (Refinery and Petrochemical Modeling System): a system description. *Bonner and Moore Management Science* 1979.

20. Borrelli, F., Bemporad, A., Morari, M. Geometric approach for multiparametric linear programming. *J. Optimization Theroy and Applications* 2003, 118, 515.

21. Castro, P., Barbosa-Povoa, A.P.F.D., Matos, H. An improved RTN continuous-time formulation for the short-term scheduling of multipurpose batch plants. *Ind. Eng. Chem. Res.* 2001, 40, 2059.

22. Cheng, L., Subramanian, E., Westerbeger, A.W. Design and planning under uncertainty: issues on problem formulation and solution. *Comput. Chem. Eng.* 2003, 27, 781.

23. Ciric, G., Gruhn, G. Tank farm management-scheduling of product movements. *Chem. Eng. Technol.* 2002, 25, 189.

24. Crema, A. The multiparametric 0-1 integer linear programming problem: a unified approach. *Eur. J. Oper. Res.* 2002, 139, 511.

25. Daniels, R.L., Carrillo, J.E. β-robust scheduling for single-machine systems with uncertain processing times. *IIE Trans.* 1997, 29, 977.

26. Das, I., Dennis, J.E. Normal boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM J. on Optimizatoin* 1998, 8, 631.

27. Dawande, M.W., Hooker, J.N. Inference-based sensitivity analysis for mixed integer/linear programming. *Oper. Res.* 2000, 48, 623.

28. Dembo, R. Scenario Optimization. *Ann. Oper. Res.* 1991, 30, 63.

29. DeWitt, C.W., Lasdon, L.S., Waren, A.D., Brenner, D.A., Melhem, S.A. Omega: an improved gasoline blending system for Texaco. *Interfaces* 1989, 19, 85.

30. Dujkanovic, M., Babic, B., Milosevic, B., Sobajic, D.J., Pao, Y.H. Fuzzy linear programming based optimal fuel scheduling incorporating blending/transloading facilities. *IEEE T. Power Syst.* 1996, 11, 1017.

31. Dua, V., Pistikopoulos, E.N. Algorithms for the solution of multiparametric mixed integer nonlinear optimization problems. *Ind. Eng. Chem. Res.* 1999, 38, 3976.

32. Dua, V., Pistikopoulos, E.N. An algorithm for the solution of multiparametric mixed integer linear programming problems. *Ann. Oper. Res.* 2000, 99, 123.

33. T. Edmunds, Bard, J.F. Algorithms for nonlinear bilevel mathematical programs. *IEEE Transactions on Systems, Man and Cybernetics* 1991, 21, 83.

34. Escudero, L.F., Quintana, F.J., Salmeron, J. CORO, a modeling and an algorithmic framework for oil supply, transformation and distribution optimization under uncertainty. *Eur. J. Oper. Res.* 1999, 114, 638.

35. Floudas, C.A. Nonlinear and mixed-integer optimization. *Oxford, New York* 1995.

36. Fortemps, P. Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Trans. on Fuzzy Systems* 1997, 5, 557.

37. Fortuny-Amat, J., McCarl, B. A representation and economic interpretation of a two-level programming problem. *Journal of Operational Research Society* 1981, 32, 783.

38. Gal, T., Nedoma J. Multiparametric linear programming. *Management Sci.* 1972, 18, 406.

39. Geoffrion, A.M. Lagrangean relaxation and its uses in integer programming. *Math. Program. Stud.* 1974, 2, 82.

40. Geoffrion, A.M., Nauss, R. Parametric and postoptimality analysis in integer linear programming. *Management Sci.* 1977, 23, 453.

41. Glismann, K., Gruhn, G. Short-term planning of blending processes: scheduling and nonlinear optimization of recipes. *Chem. Eng. Tech.* 2001, 24, 246.

42. Glismann, K., Gruhn, G. Short-term scheduling and recipe optimization of blending processes. *Comput. Chem. Eng.* 2001, 25, 627.

43. Gothe-Lundgren, M., Lundgren, J.T., Persson, J.A. An optimization model for refinery production scheduling. *Int. J. Prod. Econ.* 2002, 78, 255.

44. Guignard, M., Kim, S. Lagrangean decomposition: a model yielding stronger Lagrangean bounds. *Math. Program.* 1987, 39, 215.

45. Gumus, Z.H., Floudas, C.A. Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization* 2001, 20, 1.

46. Haimes, Y.Y. Integrated system identification and optimization. *Control and Dynamic Systems: Advances in Theory and Applications* 1973,10,435.

47. Hansen, P., Jaumard, B., Savard, G. New branch-and-bound rules for linear bilevel programming. *SIAM Journal of Scientific and Statistical Computing* 1992, 13, 1194.

48. Hillermeier, C. Generalized homotopy approach to multiobjective optimization. *J. Optimization Theory and Applications* 1995, 19, 1089.

49. Holm, S., Klein, D. Three methods for post-optimal analysis in integer linear programming. *Math. Prog. Study* 1984, 21, 97.

50. Horst, R., Tuy, H. Global optimization: deterministic approaches. *2nd Edn. Springer-Verlag, Berlin* 1993.

51. Ierapetritou, M.G.,Floudas, C.A. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Ind. Eng. Chem. Res.* 1998, 37, 4341.

52. Ierapetritou, M.G.,Floudas, C.A. Effective continuous-time formulation for short-term scheduling. 2. Continuous and semi-continuous processes. *Ind. Eng. Chem. Res.* 1998, 37, 4360.

53. Ishibuchi, H., Yamamoto, N., Murata, T., Tanaka, H. Genetic algorithms and neighborhood search algorithms for fuzzy flowshop scheduling problems. *Fuzzy Set. Syst.* 1994, 67, 81.

54. Ishizuka, Y., Aiyoshi, E. Double penalty method for bilevel optimization problems. *Annals of Operations Research* 1992, 34, 73.

55. Jenkins, L. Parametric methods in integer linear programming. *Ann. Oper. Res.* 1990, 27, 77.

56. Jenkins, L., Peters, D. A computational comparison of Gomory and knapsack cuts. *Compt. Oper. Res.* 1987, 14, 449.

57. Jia, Z., Ierapetritou, M.G., Kelly, J.D. Refinery short-term scheduling using continuous time formulation: crude oil operations. *Ind. Eng. Chem. Res.* 2003, 42, 3085.

58. Jia, Z., Ierapetritou, M.G. Short-tem scheduling under uncertainty using MILP sensitivity analysis. *Ind. Eng. Chem. Res.* 2004, 43, 3782.

59. Jia, Z., Ierapetritou, M.G. Short-tem scheduling under uncertainty using MILP sensitivity analysis. *Ind. Eng. Chem. Res.* 2004, 43, 3782.

60. Joly M., Moro L.F.L., Pinto J.M. Planning and scheduling for petroleum refineries using mathematical programming. *Braz. J Chem. Eng.* 2002, 19, 27.

61. Kelly, J.D., Forbes, J. Structured approach to storage allocation for improved process controllability. *AIChE* 1998, 44, 8, 1832.

62. Kolstad, C.D., Lasdon, L.S. Derivative estimation and computational experience with large bilevel mathematical programs. *Journal of Optimization Theory and Applications* 1990, 65, 485.

63. Kondili, E., Pantelides, C.C., Sargent, R.W.H. A general algorithm for scheduling batch operations. *Comput. Chem. Eng.* 1993, 17, 211.

64. Lababidi H.M.S., Kotob S., Yousuf B. Refinery advanced process control planning system. *Comput. Chem. Eng.* 2002, 26, 1303.

65. Lee, H., Pinto, J.M., Grossmann, I.E., Park, S. Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. *Ind. Eng. Chem. Res.* 1996, 35, 1630.

66. Li, W.K., Hui, C.W., Hua, B., Tong, Z.X. Scheduling crude oil unloading, storage and processing. *Ind. Eng. Chem. Res.* 2002, 41, 6723.

67. Li, W.K., Hui, C.W., Li, P., Li, A.X. Refinery planning under uncertainty. *Ind. Eng. Chem. Res.* 2002, 43, 6742.

68. Lin, X., Janak, S.L., Floudas, C.A. A new robust optimizaton approach for scheduling under uncertainty: I. Bounded Uncertainty. *Comput. Chem. Eng.* 2004, 28, 1069.

69. Liu, M.L., Sahinidis, N.V. Process planning in a fuzzy enviroment. *Eur. J. Oper. Res.* 1997, 100, 142.

70. Maravelias, C.T., Grossmann, I.E. New Continuous-Time State Task Network Formulation for the Scheduling of Multipurpose Batch Plants. *Ind. Eng. Chem. Res.* 2003, 42, 3056.

71. Marsten, R.E., Morin, T.L. Parametric integer programming: the right-hand side case. *Ann. Discr. Math.* 1977, 1, 375.

72. Mockus, L., Reklaitis, G.V. Continuous time representation approach to batch and continuous process scheduling. 1. MINLP formulation. *Ind. Eng. Chem. Res.* 1999, 38, 1973.

73. Moro, L.F.L., Zanin, A.C., Pinto, J.M. A planning model for refinery diesel production. *Comput. Chem. Eng.* 1998, 22, suppl., S1039.

74. Moro, L.F.L. Process technology in the petroleum refining industry - current situation and future trends. *Comput. Chem. Eng.* 2003, 27, 1303.

75. Mulvey, J.M.,Vanderbei, R.J.,Zenios, S.A. Robust optimization of large-scale systems. *Oper. Res.* 1995, 43, 264.

76. Neiro, S.M.S., Pinto, J.M. Multiperiod optimization for production planning of petroleum refineries. *Chem. Eng. Commun.* 2005, 192, 62.

77. Ohtake, Y., Nishida, N. A branch-and-bound algorithm for 0-1 parametric mixed integer programming. *Oper. Res. Lett.* 1985, 4, 41.

78. Orcun, S.,Altinel, I.K., Hortacsu, O.J. Scheduling of batch processes with operational uncertainties. *Comp. Chem. Eng.* 1996, 20, S1191.

79. Paolucci, M., Sacile, R., Boccalatte, A. Allocating crude oil supply to port and refinery tanks: a simulation-based decision support system. *Decis. Support. Syst.* 2002, 33, 39.

80. Pertsinidis, A., Grossmann, I.E., McRae, G.J. Parametric optimization of MILP programs and a framework for the parametric optimization of MINLPs. *Comp. Chem. Eng.* 1998, 22, S205.

81. Petkov, S.B.,Maranas, C.D. Multiperiod planning and scheduling of multiproduct batch plants under demand uncertainty. *Ind. Eng. Chem. Res.* 1997, 36, 4864.

82. Pinto, J.M., Grossmann, I.E. Optimal cyclic scheduling of multistage continuous multiproduct plants. *Comp. Chem. Eng.* 1994, 18, 9, 797.

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

84. Pinto, J.M., Joly, M., Moro, L.F.L. Planning and scheduling models for refinery operations. *Comp. Chem. Eng.* 2000, 24, 2259.

85. Piper, C.J., Zoltners, A.A. Some easy postoptimality analysis for zero-one programming. *Management Sci.* 1976, 22, 759.

86. Pistikopoulos, E.N., Ierapetritou, M.G. A novel approach for optimal process design under uncertainty. *Comp. Chem. Eng.* 1995, 19, 1089.

87. Quesada, I., Grossmann, I.E. Global optimization of bilinear process network with multicomponent flows. *Comput. Chem. Eng.* 1995, 19, 1219.

88. Ravi, V., Reddy, P.J. Fuzzy linear frractional goal programming applied to refinery operations planning. *Fuzzy Set. Syst.* 1998, 96, 173.

89. Rejowski, R., Pinto, J.M. Scheduling of a multiproduct pipeline system. *Comput. Chem. Eng.* 2003, 27, 1229.

90. Reklaitis, G.V. Overview of scheduling and planning of batch process opterations. *NATO Advanced Study Institute* 1992, Antalya, Turkey.

91. Rigby, B., Lasdon, L.S., Waren, A.D. The evolution of Texaco blending systems - from omega to starblend. *Interfaces* 1995, 25, 64.

92. Roodman, G.M. Postoptimality analysis in zero-one programming by implicit enumeration. *Naval Res. Log. Quarterly* 1972, 19, 435.

93. Roodman, G.M. Postoptimality analysis in integer programming by implicit enumeration: the mixed integer case. *Naval Res. Log. Quarterly* 1974, 21, 595.

94. Ryoo, H.S., Sahinidis, N.V. Global optimization of nonconvex NLPs and MINLPs with applications in process design. *Comput. Chem. Eng.* 1995, 19, 551.

95. Sahinidis, N.V. BARON: A general purpose global optimization software package. *J. Global Optimization.* 1996, 8, 201.

96. Sahinidis, N.V. Optimization under uncertainty: state-of-the-art and opportunities. *Comput. Chem. Eng.* 2004, 28, 971.

97. Sahinidis, N.V., Grossmann, I.E. MINLP model for cyclic multiproduct scheduling on continuous parallel lines. *Comput. Chem. Eng.* 1991, 15, 85.

98. Samsatli, N.J.,Papageorgiou, L.G.,Shah, N. Robustness metrics for dynamic optimization models under parameter uncertainty. *AICHE J* 1998, 44, 1993.

99. Savard, G., Gauvin, J. The steepest descent direction for the nonlinear bilevel programming problem. *Operations Research Letters* 1994, 15, 265.

100. Schilling, G., Pantelides, C.C. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Comput. Chem. Eng.* 1996, 20, S1221.

101. Schmidt, C., Grossmann, I.E. Optimization models for the scheduling of testing tasks in new product development. *Ind. Eng. Chem. Res.* 1996, 35, 3498.

102. Schrage, L., Wolsey, L. Sensitivity analysis for branch and bound integer programming. *Oper. Res.* 1985, 33, 1008.

103. Shah, N. Mathematical programming techinques for crude oil scheduling. *Comput. Chem. Eng.* 1996, 20, suppl., S1227.

104. Shah, N., Pantelides, C.C. Design of multipurpose batch plants with uncertain production requirements. *Ind. Eng. Chem. Res.* 1992, 31, 1325.

105. Singh, A., Forbes, J.F., Vermeer, P.J., Woo, S.S. Model-based real-time optimization of automotive gasoline blending operations. *J. Process Contr.* 2000, 10, 43.

106. Skorin-Kapov, J., Granot, F. Non-linear integer programming: sensitivity analysis for branch and bound. *Oper. Res. Letters* 1987, 6, 269.

107. Song J., Park H., Lee D.Y. Scheduling of actual size refinery processes considering environmental impacts with multiobjective optimization. *Ind. Eng. Chem. Res.* 2002, 41, 4794.

108. Steinschorn, D., Hofferl, F. Refinery scheduling using mixed integer LP and dynamic recursion. *NPRA Computer Conference, New Orleans* 1997.

109. Straub, D.A., Grossmann, I.E. Design optimization of stochastic flexibility. *Comput. Chem. Eng.* 1993, 17, 339.

110. Vicente, L.N., Calamai, P.H. Bilevel and multilevel programming: a bibliography review. *Journal of Global Optimization* 1994, 6, 291.

111. Vin, J.P.,Ierapetritou, M.G. Robust short-term scheduling of multiproduct batch plants under demand uncertainty. *Ind. Eng. Chem. Res.* 2001, 40, 4543.

112. Wu, D., Ierapetritou, M.G. Decomposition approaches for the efficient solution of short-term scheduling problems. *Comput. Chem. Eng.* 2003, 27, 1261.

113. Zadeh, L. Optimality and nonscalar-valued performance criteria. *IEEE Transactions on Automatic Control* 1963,8,59.

114. Zhang, N., Zhu, X.X. A novel modelling and decomposition strategy for overall refinery optimisation. *Comput. Chem. Eng.* 2000, 24, 1543.

115. Zhang, Y., Monder, D., Forbes, J.F. Real-time optimization under parametric uncertainty: a probability constrained approach. *J. Process Contr.* 2002, 12, 373.

Zhenya Jia

Curriculum Vitae

Education

2000-2005 **PhD, Rutgers, The State University of New Jersey**

1994-1999 **BS, Tsinghua University, Beijing**

Publications

- Jia, Z. and M.G. Ierapetritou, Uncertainty analysis on the right-hand-side for MILP problems, *submitted for publication, AICHE Journal,* 2005.
- Jia, Z. and M.G. Ierapetritou, Generate pareto optimal solutions of scheduling problems using normal boundary intersection technique, *submitted for publication, Computers and Chemical Engineering,* 2005.
- Jia, Z. and M.G. Ierapetritou, Refinery scheduling considering demand uncertainty, *manuscript in preparation* 2005.
- K.C.Furman, Jia, Z., and M.G. Ierapetritou, Continuous time formulation for tank transfer scheduling, *manuscript in preparation,* 2005.
- Jia, Z. and M.G. Ierapetritou, Efficient short-term scheduling of refinery operations based on a continuous time formulation, *Computers and Chemical Engineering*, 28, 1001, 2004.
- Jia, Z. and M.G. Ierapetritou, Short-term scheduling under uncertainty using MILP sensitivity analysis, *Industrial and Engineering Chemistry Research*, 43, 3782, 2004.
- Jia, Z., M.G. Ierapetritou, and J. D. Kelly, Refinery Short-term scheduling using continuous time formulation - crude oil operations, *Industrial and Engineering Chemistry Research*, 42, 3085, 2003.
- Jia, Z. and M.G. Ierapetritou, Mixed-integer programming model for gasoline blending and distribution scheduling, *Industrial and Engineering Chemistry Research*, 42, 825, 2003.