DECOMPOSITION APPROACHES FOR ENTERPRISE-WIDE OPTIMIZATION IN PROCESS INDUSTRY

by

NIKISHA K. SHAH

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

Marianthi G. Ierapetritou, PhD

And approved by

____

____

____

____

New Brunswick, New Jersey

May, 2015

ABSTRACT OF THE DISSERTATION

Decomposition Approaches For Enterprise-Wide Optimization In Process Industry

By NIKISHA K. SHAH

Dissertation Director:

Marianthi G. Ierapetritou, PhD

Enterprise-wide optimization (EWO) has gain lot of interest in recent years as the globalization trends of past few decades have significantly increased the scale and complexity of modern process industry and increasing economic pressures to remain competitive in global marketplace. EWO entails optimization of supply, manufacturing, and distribution activities to reduce costs and inventories through an integrated and coordinated decision-making among various functions in the industry (vendors, production facilities, and distribution). One of the major challenges in achieving EWO is mathematical tools for planning and scheduling for manufacturing facilities.

Main objective of this dissertation is to develop mathematical methodologies to assist in achieving EWO goals for chemical process industry. Specially, mathematical formulation for planning and scheduling decisions and decomposition strategies will be developed in order to bridge the gap between concepts and industrial application. In this work, planning and scheduling of multisite, multiproduct batch production and distribution faculties is addressed via dual decomposition based approach, which aims to reduce computational complexity through parallel computation. In the area of continuous production facilities, lack of efficient scheduling models prevents us from developing coordinated planning and scheduling tools. To address this issue,

mathematical formulations for scheduling of refinery operations is developed and novel heuristics and mathematical decomposition strategies for large scale complex mixed integer linear programming models are proposed. Throughout this dissertation, case studies will be used to demonstrate the applicability of proposed decomposition approaches.

Acknowledgements

I would never have been able to finish my PhD without the support and encouragement from my family and my advisor. I have been extremely fortunate to have a wonderful parents, advisor, and friends supporting me for the last few years. Without all of your guidance and help, I would not have reached the finished line.

I would like to express my deepest and sincere gratitude to my advisor, Dr. Marianthi Ierapetritou, for her excellent guidance, caring, tremendous patience, encouragement, and providing me with an excellent atmosphere for doing research and grow as an independent thinker. Her mentorship was paramount in helping me grow academically, professionally and personally. I am not sure many graduate students are given the opportunity to develop their own individuality or allowed to work with such independence. Thank you for always supporting me, challenging me to do my best, and believing in me when I had lost faith in myself. For everything you've have done for me, Dr. Ierapetritou, I thank you.

I would also like to thank the members of my committee, Dr. Ioannis P. Androulakis, Dr. Rohit Ramachandran, and Dr. Beverly Smith for their support and kind accommodations.

I was fortunate to have Dr. Zhenya Jia introduce me to my research topic when I was still an undergraduate. I would especially like to thank Dr. Georgios Saharidis for his assistance and guidance in getting my interest in decomposition strategies started on the right foot and providing me with honest critical feedback on my first research article which has helped greatly improved my ability to communicate my research effectively.

I wish to acknowledge past and present members of in the Laboratory for Optimization and Systems Analysis and friends at Rutgers whose friendship and support I will always treasure: Dr. Edgar Davis, Dr. Patricia Portillo, Dr. Hong Yang, Dr. Zukui Li, Dr. Vidya Iyer, Dr. Mehmet Orman, Dr. Aditya Vanarase, Dr. Fani Boukouvala, Dr. Yijie Gao, Dr. Kaiyuan He, Dr. Shuliang

Zhang, Dr. Ravendra Singh, Dr. Amanda Rogers, Dr. Jinjun Zhuge, Nihar Sahay, Zhaojia Lin, Parham Farzan, and Sebastian Escotet.

Lastly, I would like to thank my family for all their love, encouragement, and unconditional support. To my parents who raised me with a love of science, engineering, and critical thinking. Without inspiring example of my mother, her friendship, and her constant belief in me that I can achieve anything through hard work and dedication, I would never have made it this far. To my sister Khushbu, for teaching me the virtues of personal strength and perseverance – in the face of difficulties. To my younger brothers Vaibhav and Dhruv, for always providing me with an entertainment.

Dedication

To my mother: Shobhana Shah

Table of Contents

List of Tables

List of Illustrations

Chapter 1

1. Introduction

1.1. Enterprise Wide Optimization

The process industry is a key sector in the global economy, converting raw materials such as crude oil, water, and natural resources into thousands of products. According to the American Chemistry Council, over 96% of all manufactured goods are dependent on chemical industry and U.S. produces over 15% of world's chemical output, amounting to US$812 billion in (2014). In particular, the petroleum refining industry is the largest source of energy products in the world and is supplying about 39% of total U.S. energy demand and 97% of transportation fuels. Process industry has grown increasingly complex in the last 20 years as a result of tighter competition, stricter environmental regulations, uncertainty in the prices of energy, raw materials, and products, and lower-margin profits. Globalization trends of past few decades have significantly increased the scale and complexity of the modern enterprise by transforming them into global network consisting of multiple business units and functions. Today process industries involve multipurpose, multisite production facilities producing hundreds of products, located in different regions and countries and servicing international market. (Wassick, 2009) In last decades, enterprises are realizing the importance of enterprise wide optimization in reducing the overall costs and remaining competitive in a dynamic global marketplace.

Enterprise-wide optimization involves coordinated optimization of the operations of supply, manufacturing, and distribution; and integration of the information and decisions-making among the various functions that comprise the supply chain of the company. (I. Grossmann, 2005; I. E. Grossmann, 2014) The concept of enterprise-wide optimization (EWO) lies at the interface of the process system engineering, operations research and supply chain optimization and these concepts are suitably positioned to provide decisions making models and algorithms for optimization of an integrated manufacturing and distribution complexes. Supply chain optimization can be considered an equivalent term for describing the enterprise-wide optimization although supply chain optimization places more emphasis on logistics and distribution, whereas enterprise-wide optimization is aimed at manufacturing facilities optimization. (Shapiro, 2001) The process industry supply chains vary from the petroleum supply chain (N. K. Shah et al., 2011) to pharmaceutical industry supply chain (Nilay Shah, 2004); however they all include manufacturing as a major component according to I. E. Grossmann (2014); Wassick (2009). The main goal of EWO is to maximize profits while minimizing costs, inventories and to this ends, major operational activities of EWO is planning, scheduling and real-time optimization and control (Figure 1.1).

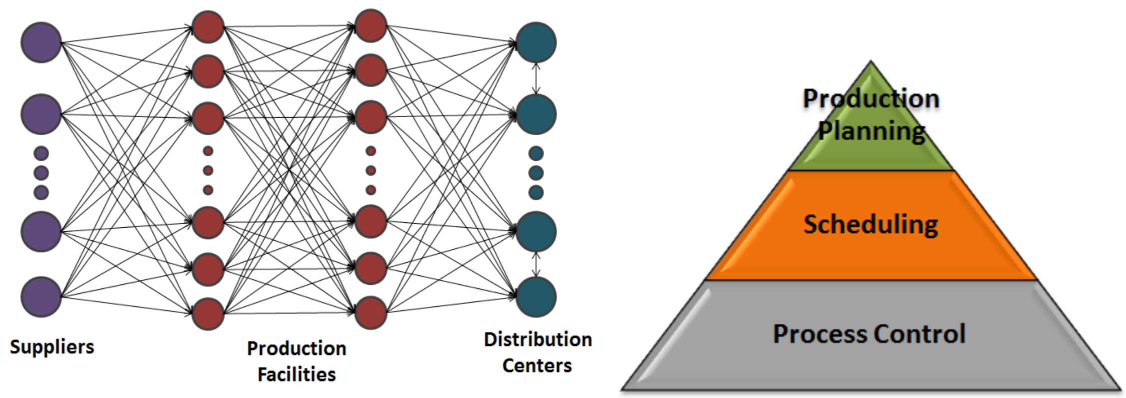

Figure 1.1 Major elements of enterprise-wide optimization.

In the field of enterprise-wide optimization, planning and scheduling are the most important operational decisions. Objectives of the planning and scheduling problems is to determine the allocation of available resources over time to perform a set of tasks required to manufacture one or more products as to satisfy global demand. The long to medium term planning covers a time horizon of between few months to a year and is concerned with decisions such as production, inventory, and distribution profiles. Short-term scheduling decision deals with issues such as assignment of tasks to units and sequencing of tasks in each unit and typically covers time horizon of between days to a few weeks. Typically, much of the decision-making in a supply chain is focused across solving sub-problems as an entity, but from the enterprise-wide performance viewpoint, local improvements at any sublevel do not necessary lead to an overall improvement and to realize full potential of EWO, integrated approach is necessary. Amongst the challenge involve in EWO in process industry, the chief challenge is coordinated decision-making across different functions in industry (procurement, manufacturing, distributions), between various geographically distributed manufacturing sites, and across three levels of decisions-making process. (Shapiro, 2001) While the first challenge relates to spatial integration and the third challenge involves temporal integration across different timescales. Coordinated decision making between geographically distributed integrated manufacturing sites deals with not only spatial integration but also with temporal integrations.

Several of the aforementioned issues can be addressed in part through integration of planning and scheduling decision-making for multi-product, multi-site production and distribution facilities.

1.1.1. Problems and Challenges

I. Grossmann (2005) discussed four major challenges in application of EWO: (a) Mathematical modeling, (b) multiscale optimization, (c) optimization under uncertainty, and (d) algorithmic and computational challenge. The first challenge involves development of production and scheduling models that can effectively capture the complexity of various operations but can also be solved in reasonable time frame. Second challenge involves difficulty associated with coordination between different time scale and over different geographically located sites. Uncertainty is inherent in supply chain (e.g. prices, demand, equipment breakdown) and effectively addressing it can effect industry profits. However, before addressing uncertainties, computationally effective deterministic models should be developed. Various models developed in previous three points, are large scale complex MILP or MINLP models and they require the application of various decomposition techniques. Comprehensive reviews of challenges faced in EWO of process industry are presented by I. E. Grossmann (2014); Nilay Shah (2004, 2005).

Scheduling models in process industry address continuous, batch, and semi-continuous production systems and complexity involved modeling these three different production system vary. In continuous production units, there is a simultaneous inlet and outlet streams, where as in batch process, simultaneous inlet and outlet streams are not allowed. Large varieties of products are produced using batch processes and food, beverages, pharmaceutical products, paint, fertilizer, and cement are a few of the categories of products produced using batch processes. Oil refinery is one of the prominent systems with continuous production process. In last ten years, great progress has been in short-term batch scheduling process, few of these works are Burkard et al. (2002); Pedro M. Castro et al. (2009); P. M. Castro et al. (2011); He and Hui (2007); Ierapetritou and Floudas (1998); Janak et al. (2006); Janak et al. (2004); Kondili et al. (1993); Maravelias and Grossmann (2003); Moniz et al. (2014). The challenge in batch process industry lies in developing effective solution methodology for integrating existing scheduling models with planning level model and integrating decisions making across multisite production and distribution facilities. Whereas, the main challenge in continuous process industry lies in developing general purpose scheduling models. Every continuous process industry has its own unique problems and arriving at one scheduling model for different types of continuous process operations is difficult and thus focus is on development of general scheduling models for specific industry. (I. Grossmann, 2005) In petroleum industry, different operations in refinery have their own scheduling model instead of an integrated scheduling model for overall refinery operations. Before tackling planning and scheduling integration in refinery, efficient and effective scheduling model for overall efficient refinery operations need to be developed. Furthermore, refinery process is very complex and their scheduling models are large scale complex mixed integer models that require novel decomposition strategies.(Kelly & Zyngier, 2008; Shaik & Floudas, 2007)

1.2. Planning and Scheduling in Process Industry

In the recent years, the area of integrated planning and scheduling has received much attention for single-site batch production facilities. However, the current manufacturing environment for process industry has changed from a traditional single-site production plant to a more integrated global production serving the emerging market. (Wassick, 2009) Modern process industries operate as a large integrated complex that involve multi-product, multi-purpose, and multi-site facilities serving a global market. The process industries supply chain can be defined to be composed of production facilities and distribution centers, where final products produced at production facilities are transported to distribution center to satisfy the customers demand. In current global market, spatially distributed production facilities across various geographical locations can no longer be regarded as isolated from each other and interactions between the production facilities and the distribution centers should be taken into account when making decisions. In this context, the issues of enterprise planning and coordination across production plants and distribution facilities are important for robust response to global demand and to maintain business competiveness, sustainability, and growth. (Papageorgiou, 2009) As the pressure to reduce the costs and inventories increases, centralized approaches have become the main policies to address supply chain optimization. (Grossmann, 2005) The integrated planning and scheduling model for multi-site facilities is important to ensure the consistency between planning and scheduling level decisions and to optimize production and transportation costs.

Wassick (2009) proposed a planning and scheduling model based on resource task network for an integrated chemical complex. He considered the enterprise-wide optimization of the liquid waste treatment network with their model. Kreipl and Pinedo (2004) discussed issues present in modeling the planning and scheduling decisions for supply chain management. For a multisite facilities, the size and level of interdependences between these sites present unique challenges to the integrated tactical production planning and day-to-day scheduling problem and these challenges are highlighted by Kallrath (2002b). For further elucidation of various aspects of planning, reader is directed to the work of Timpe and Kallrath (2000) and Kallrath (2002a).

1.3. Refinery Operations Scheduling

The continuous process plants scheduling have drawn less consideration in the literature compared to that of batch plants even though continuous plants are prevalent in the chemical process industries. Of the continuous process industries, the oil refinery production operation is one of the most complex chemical industries, which involves many different and complicated processes with various connections. Instead of tackling a comprehensive large-scale refinery operations optimization problem, decomposition approaches are generally exploited. Oil refinery manufacturing operations can be decomposed into three problems: (1) crude-oil unloading, mixing, and inventory control, (2) scheduling of production units, and (3) finish products blending and distribution. (Jia & Ierapetritou, 2004) The goal of EWO is to optimized overall refinery operations in a coordinated fashioned. Depending on the problem characteristics as well as the required flexibility in the solution, scheduling models can be based on either a discrete, a continuous, or hybrid time domain representation. (Iiro Harjunkoski et al., 2014; J. Li & Karimi, 2011; Mouret et al., 2010; Neiro et al., 2014) Real-life features such as multipurpose production units, multipurpose product tanks, parallel non-identical blenders, minimum run lengths, sequence dependent changeovers, product giveaway, piecewise constant profiles for blend component qualities and feed rates, etc. introduces in more operational constraints and many combinatorial decisions, that renders this large scale mixed integer problem difficult to solve without decomposition solution strategies. (Kelly, 2006; Kelly & Zyngier, 2008; Shaik & Floudas, 2007; Shaik et al., 2009)

1.4. Motivation: development and implementation of decomposition tools for planning and scheduling in process industry

Despite the many of the potential benefits in coordinated decisions making in EWO, aforementioned challenges in coordinated decisions making in planning and scheduling for multiproduct, multisite product and distribution facilities and scheduling of overall refinery operations, concepts of enterprise-wide optimization are underutilized. This is due to the difficulties associated with building effective mathematical formulation that captures real world complexity without becoming incomprehensive to solve and for efficient models to be utilized for real world application, the challenge lies in developing decomposition approaches for large scale mixed integer models.(I. E. Grossmann, 2014)

First objective of this dissertation is to develop mathematical formulation for multisite batch-process production and distribution facility planning and propose decomposition approach to address multi-scale optimization problem arising from integration of planning and scheduling decisions level. Second objective of this dissertation is to propose scheduling models for large scale refinery operations, to develop efficient decomposition based methodology to address large scale optimization problems and demonstrate applicability of decomposition methodologies. The methodologies to tackle large scale optimization models include, valid inequalities, heuristic algorithm, Lagrangian relaxation decomposition, and augmented Lagrangian optimization. Each of these decomposition methods will be discussed in the context of refinery operations scheduling and demonstrated using case studies related to real refinery complex.

1.5. Outline of dissertation

Each of the six main chapters in this dissertation will emphasize a specific concept or tool for enterprise-wide optimization. Chapter 2 will present decomposition based methodology to solve full space integrated planning and scheduling problem for multisite, multiproduct batch production plants and multisite distribution facilities. Chapter 3 will present spatial decomposition strategy for refinery operations where centralized and decentralized decision making process is compared and conditions during which decentralized approach gives global optimal solution are discussed. A unified comprehensive refinery operation scheduling model that incorporates many of the real world operational and logistics rules is presented in Chapter 4. The resulting model is large scale mixed integer linear programming model (MILP) and valid inequalities are developed to reduce complexity of the model by reducing the total number of nodes and iterations in branch and bound framework. Refinery operations scheduling model incorporating many logistics rules is difficult to solve to optimality in reasonable computational time even after inclusion of valid inequalities. For refinery without any blend component tanks, Chapter 5 tackles complexity of large scale scheduling model by developing mathematical decomposition, Lagrangian relaxation, while Chapter 6 focuses on efficient heuristic decomposition algorithm. In Chapter 7, a general augmented Lagrangian decomposition for different refinery operations configuration (rundown streams, blend component tanks, or both) is introduced and applied to number of case studies to illustrate its effectiveness.

Chapter 2

2. Planning and scheduling of multisite batch production and distribution facilities

The current manufacturing environment for process industry has changed from a traditional single-site, single market to a more integrated global production mode where multiple sites are serving a global market. In this chapter, the integrated planning and scheduling problem for the multisite, multiproduct batch plants is considered. The major challenge for addressing this problem is that the corresponding optimization problem becomes computationally intractable as the number of production sites, markets, and products increases in the supply chain network. To effectively deal with the increasing complexity, the block angular structure of the constraints matrix is exploited by relaxing the inventory constraints between adjoining time periods using the augmented Lagrangian decomposition method. To resolve the issues of non-separable cross-product terms in the augmented Lagrangian function, diagonal approximation method is applied. Several examples have been studied to demonstrate that the proposed approach yields significant computational savings compared to the full-scale integrated model.

2.1. Introduction

Modern process industries operate as a large integrated complex that involve multiproduct, multipurpose, and multisite production facilities serving a global market. The process industries supply chain is composed of production facilities and distribution centers, where the final products are transported from the production facilities to distribution centers and then to retailers to satisfy the customers demand. In current global market, spatially distributed production facilities across various geographical locations can no longer be regarded as independent from each other and interactions between the manufacturing sites and the distribution centers should be taken into account when making decisions. In this context, the issues of enterprise planning and coordination across production plants and distribution facilities are important for robust response to global demand and to maintain business competeveness, sustainability, and growth.

(Papageorgiou, 2009) As the pressure to reduce the costs and inventories increases, centralized approaches have become the main policies to address supply chain optimization. An excellent overview of the enterprise-wide optimization (EWO) and the challenges related to process industry supply chain is highlighted by I. Grossmann (2005). Varma et al. (2007) described the main concepts of EWO and presented the potential research opportunities in addressing the problem of EWO models and solution approaches.

Supply chain optimization can be considered an equivalent term for describing the enterprise-wide optimization according to Shapiro (2001) although supply chain optimization places more emphasis on logistics and distribution, whereas enterprise-wide optimization is aimed at manufacturing facilities optimization. Key issues and challenges faced by process industry supply chain are highlighted by Nilay Shah (2004, 2005). Traditional supply chain management planning decisions can be divided into three levels: strategic (long-term), tactical (medium-term), and operational (short-term). The long-term planning determines the infrastructure (e.g. facility location, transportation network). The medium-term planning covers a time horizon between few months to a year and is concerned with decisions such as production, inventory, and distribution profiles. Finally, short-term planning decision deals with issues such as assignment of tasks to units and sequencing of tasks in each unit. The short-term planning level covers time horizon between days to a few weeks and at production level, is typically refer to as scheduling. Wassick (2009) proposed a planning and scheduling model based on resource task network for an integrated chemical complex. He considered the enterprise-wide optimization of the liquid waste treatment network with their model. Kreipl and Pinedo (2004) discussed issues present in modeling the planning and scheduling decisions for supply chain management. For a multisite facilities, the size and level of interdependences between these sites present unique challenges to the integrated tactical production planning and day-to-day scheduling problem and these challenges are highlighted by Kallrath (2002b). For further elucidation of various aspects of planning, the reader is directed to the work of Timpe and Kallrath (2000) and Kallrath (2002a).

A simple network featuring the multisite facilities is given in Figure 2.1, where multiple products may be produced in individual process plants at different locations spread across geographic region and then transported to distribution centers to satisfy customers demand. These multisite plants produce a number of products driven by market demand under operating conditions such as sequence dependent switchovers and resource constraints. Each plant within the enterprise may have different production capacity and costs, different product recipes, and different transportation costs to the markets according to the location of the plants. To maintain economic competitiveness in a global market, interdependences between the different plants, including intermediate products and shared resources need be taken into consideration when making planning decisions. Furthermore, the planning model should take into account not only individual production facilities constraints but also transportation constraints because in addition to minimizing the production cost, it’s important to minimize the costs of products transportation from production facilities to the distribution center. Thus, simultaneous planning of all activities from production to distribution stage is important in a multisite process industry supply chain. (N. Shah, 1998)

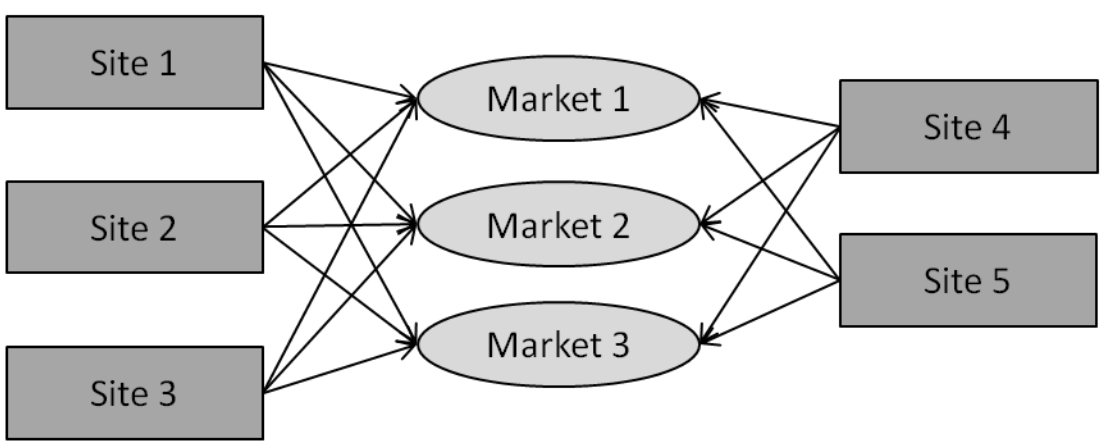

Figure 2.1 Multisite production and distribution network

Wilkinson et al. (1996) proposed an aggregated planning model based on the resource task network framework developed by Pantelides (1994). Their proposed planning model considers integration of production, inventory, and distribution in multisite facilities. Lin and Chen (2007) developed a multistage, multisite planning model that deals with routings of manufactured products demand among different production plants. They simultaneously combine two different time scales (i.e. monthly and daily) in their formulation by considering varying time buckets. Verderame and Floudas (2009) developed an operational planning model which captures the interactions between production facilities and distribution centers in multisite production facilities network. Their proposed multisite planning with product aggregation model (Multisite-PPDM) incorporates a tight upper bound on the production capacity and transportation cost between production facilities and customers distribution centers in the supply chain network under consideration. A multisite production planning and distribution model is proposed by Jackson and Grossmann (2003) where they utilized nonlinear process models to represent production facilities. They have exploited two different decomposition schemes to solve the large-scale nonlinear model using Lagrangian decomposition. In temporal decomposition, the inventory constraints between adjoining time periods are dualized in order to optimize the entire network for each planning time period. In spatial decomposition technique, interconnection constraints between the sites and markets are dualized in order to optimize each facility individually. They conclude that temporal decomposition technique performs far better than spatial decomposition technique.

The traditional strategy to address planning and scheduling level decisions is to follow a hierarchical approach in which planning decisions are made first and then scheduling decisions are made using planning demand targets. However, this approach does not consider any interactions between the two decision making levels and thus the planning decisions may result in suboptimal or even infeasible scheduling problems. Due to significant interactions between planning and scheduling decisions levels in order to determine the global optimal solution it is necessary to consider the simultaneous optimization of the planning and scheduling decisions. However, this simultaneous optimization problem leads to a large problem size and the model becomes intractable when typical planning horizon is considered. For an overview of issues, challenges and optimization opportunities present in production planning and scheduling problem, the reader is referred to the work of Maravelias and Sung (2009).

In recent years, the area of integrated planning and scheduling for single site has received much attention. Different decomposition strategies are developed to effectively deal with a large scale integrated model. One of the existing approaches follows a hierarchical decomposition method, where the upper level planning problem provides a set of decisions such as production and inventory targets to the lower level problem to determine the detailed schedule. If the solution of lower level problem is infeasible, an iterative framework is used to obtain a feasible solution. (Bassett et al., 1996) To further improve this approach, tight upper bounds on production capacity are implemented in upper level problem in presence of an approximate scheduling model or aggregated capacity constraints. (Nilay Shah, 2005; Shapiro, 2001) Another related idea is the one that follows a hierarchical decomposition within a rolling horizon framework. In this model detailed scheduling models are used for a few early periods and aggregated models are used for later periods. (Dimitriadis et al., 1997; Z. Li & Ierapetritou, 2010b; Verderame & Floudas, 2008; Wu & Ierapetritou, 2007) A different decomposition strategy is based on the special structure of the large-scale mathematical programming model. The integrated planning and scheduling model has a block angular structure which arises when a single scheduling problem is used over multiple planning periods. The constraints matrix of the integrated problem has complicating variables that appear in multiple constraints. By making copies of the complicating variables, the complicating variables are transformed into complicating constraints (linking constraints) and these complicating constraints can be relaxed using the Lagrangian relaxation method. One major drawback of the Lagrangian relaxation (LR) is that there is duality gap between the solution of the Lagrangian relaxation method and original problem and to resolve this issue, augmented Lagrangian relaxation (ALR) method should be used Y. Li et al. (2008); Tosserams et al. (2006); Tosserams et al. (2008). Z. Li and Ierapetritou (2010a) applied augmented Lagrangian optimization method to integrated planning and scheduling problem for single site plants. One disadvantage of ALR method is the non-separability of the relaxed problem which arises due to the quadratic penalty terms present in the objective function. To resolve the issue of the non-separability, Z. Li and Ierapetritou (2010a) studied two different approaches. The first approach is based on linearization the cross-product terms using diagonal quadratic approximation (DQA). (Y. Li et al., 2008) However, in this approach, an approximation of the relaxation problem is solved and it may not lead to a global optimal solution of the original problem. In the second approach, Z. Li and Ierapetritou (2010a) proposed a two-level optimization method which solves an exact relaxation problem. However, the proposed two-level optimization strategy requires a non-smooth quadratic problem to be optimized at every iteration. They conclude that DQA-ALO method is more effective than the two-level optimization method for the integrated planning and scheduling problems.

Even though most companies operate in a multisite production manner, very limited attention has been paid on integrating planning and scheduling decisions for multisite facilities. The integrated planning and scheduling model for multisite facilities is important to ensure the consistency between planning and scheduling level decisions and to optimize production and transportation costs. Since the production planning and scheduling level deals with different time scales, the major challenge for the integration using mathematical programming methods lies in addressing large scale optimization models. The full-scale integrated planning and scheduling optimization model spans the entire planning horizon of interest and includes decisions regarding all the production sites and distribution centers. When typical planning horizon is considered, the integrated full-scale problem becomes intractable and a mathematical decomposition solution approach is necessary.

In this work, augmented Lagrangian relaxation method is applied to solve the multisite production and distribution optimization problem. The chapter is organized as follow. The problem statement is given in Section 2.2, whereas Section 2.3 presents the problem formulation. The general augmented Lagrangian method and its application to multisite facility is given in Section 2.4. The results of examples studied are shown in section 2.5 and the chapter concludes with summary in section 2.6.

2.2. Problem Statement

The supply chain network (Figure 2.1) under investigation contains multiple batch production facilities which supply products to multiple distribution centers. Every production site may supply all distribution centers but all the products may not be produced at every production site. In the proposed model, it is assumed that one cannot sell more products to a market than the market forecasted demand. Thus the requested demand acts as an upper bound on finished product sales. The proposed model tries to satisfy the market demands; however it allows for unsatisfied demand to be carried over to the next planning period (backorder) and also allows for partial order fulfillment. The unsatisfied demand and backorders are penalized on a daily basis in order to maximize the degree of customer demand order fulfillment. Following assumptions are made: unlimited supply of raw materials is available and fixed and variable production, storage, and backorder costs are known for the planning horizon under consideration. The transportation costs are also assumed to be known. It is further assumed that there are no shipping delays in the network and the length of time of the planning horizon is such that the effects of transportation delays are neglected.

Given the daily demand profiles for each distribution center, the goals of the integrated planning and scheduling problem is to ascertain the daily production target profiles for each production facilities and product shipment profiles from production facilities to distribution centers so that demand is satisfied over the planning horizon under considerations (several months up to a year). The objective of the integrated problem is to minimize inventory, backorder, transportation, and production costs.

2.3. Mathematical formulation

The multisite model includes production site constraints and distribution center (market) constraints. The set of products s ∈ PR are to be produced at various production sites (p ∈ PS) and are to be distributed to a global market (m ∈ M) over planning horizon (t ∈ T). To formulate the integrated planning and scheduling model, an integrated modeling approach is proposed in which planning and scheduling decisions level constraints are incorporated. The planning horizon is discretized into fixed time length (daily production periods) and for each planning period, a detailed scheduling model for each batch production facilities is considered. The detailed scheduling model is based on continuous time representation and notion of event points. (Ierapetritou & Floudas, 1998) The planning and scheduling decisions levels are inter-connected via production and inventory constraints. The integrated planning and scheduling model for a single-site proposed by Z. Li and Ierapetritou (2010a) is extended to accommodate multisite production facilities serving multiple markets. The extended integrated multisite model is as follows.

$$\begin{aligned} \min \; & \sum_t \sum_p \sum_{s \in S_f^p} h_s^p Inv_s^{p,t} + \sum_t \sum_m \sum_{s \in S_f^m} u_s^m U_s^{m,t} + \sum_t \sum_m \sum_p \sum_{s \in S_f^p \cap S_f^m} d_s^{p,m} D_s^{p,m,t} \\ & + \sum_t \sum_p \sum_{j \in J^p} \sum_{i \in I_j^p} \sum_n \left(FixCost_i^p wv_{i,j,n}^{p,t} + VarCost_i^p b_{i,j,n}^{p,t} \right) \end{aligned} \tag{1a}$$

$$s.t. \quad Inv_s^{p,t} = Inv_s^{p,t-1} + P_s^{p,t} - \sum_m D_s^{p,m,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{1b}$$

$$U_s^{m,t} = U_s^{m,t-1} + Dem_s^{m,t} - \sum_{p \in P_s} D_s^{p,m,t}, \quad \forall s \in S_f^m, m \in M, t \in T \tag{1c}$$

$$st_{s,n=N}^{p,t} - stin_s^{p,t} = P_s^{p,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{1d}$$

$$stin_s^{p,t} = Inv_s^{p,t-1}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{1e}$$

$$\sum_{i \in I_j^p} wv_{i,j,n}^{p,t} \le 1, \quad \forall j \in J^p, n \in N, p \in PS, t \in T \tag{1f}$$

$$v_{i,j,p}^{\min} wv_{i,j,n}^{p,t} \le b_{i,j,n}^{p,t} \le v_{i,j,p}^{\max} wv_{i,j,n}^{p,t}, \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1g}$$

$$st_{s,n}^{p,t} \le stcap_s^p, \quad \forall s \in S^p, n \in N, p \in PS, t \in T \tag{1h}$$

$$st_{s,n}^{p,t} = st_{s,n-1}^{p,t} - \sum_{i \in I_s^p} \rho_{s,i}^{c,p} \sum_{j \in J_i^p} b_{i,j,n}^{p,t} + \sum_{i \in I_s^p} \rho_{s,i}^{p,p} \sum_{j \in J_i^p} b_{i,j,n-1}^{p,t}, \quad \forall s \in S^p, n \in N, p \in PS, t \in T \tag{1i}$$

$$st_{s,n=1}^{p,t} = stin_s^{p,t} - \sum_{i \in I_s^p} \rho_{s,i}^{c,p} \sum_{j \in J_i^p} b_{i,j,n=1}^{p,t}, \quad \forall s \in S^p, p \in PS, t \in T \tag{1j}$$

$$Tf_{i,j,n}^{p,t} = Ts_{i,j,n}^{p,t} + \alpha_{i,j}^p wv_{i,j,n}^{p,t} + \beta_{i,j}^p b_{i,j,n}^{p,t}, \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1k}$$

$$Ts_{i,j,n+1}^{p,t} \ge Tf_{i,j,n}^{p,t} - H\left(1 - wv_{i,j,n}^{p,t}\right), \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1l}$$

$$Ts_{i,j,n+1}^{p,t} \ge Tf_{i',j,n}^{p,t} - H\left(1 - wv_{i',j,n}^{p,t}\right), \quad \forall i \in I_j^p, i' \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1m}$$

$$Ts_{i,j,n+1}^{p,t} \ge Tf_{i',j',n}^{p,t} - H\left(1 - wv_{i',j',n}^{p,t}\right), \quad \forall i, i' \in I_j^p, i' \ne i, j, j' \in J^p, n \in N, p \in PS, t \in T \tag{1n}$$

$$Ts_{i,j,n+1}^{p,t} \ge Ts_{i,j,n}^{p,t}, \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1o}$$

$$Tf_{i,j,n+1}^{p,t} \ge Tf_{i,j,n}^{p,t}, \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1p}$$

$$Ts_{i,j,n}^{p,t} \le H, \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1q}$$

$$Tf_{i,j,n}^{p,t} \le H, \quad \forall i \in I_j^p, j \in J^p, n \in N, p \in PS, t \in T \tag{1r}$$

The objective function shown in equation (1a) minimizes the total costs of the integrated model, which includes variable inventory costs, backorder costs, transportation costs, and production costs and fixed production costs. The planning level is modeled by constraints (1b-1c). Equation 1b predicts the production targets ($P_s^{p,t}$), inventory targets ($Inv_s^{p,t}$), and shipping targets ($D_s^{p,m,t}$) for each product. The constraint describing each distribution center (market) is given in equation (1c). As shown in equation (1c), the backorder balance is performed for each customer market by considering the demand forecast ($Dem_s^{m,t}$) of that market and the sales all the shipments of the product from one or combination of all production sites to that market ($D_s^{p,m,t}$). Constraints (1d) assign production targets ($P_s^{p,t}$) of planning level solutions to scheduling level problem for each product to each production facility for different planning periods. Equation (1e) represents the connection between the inventory level requirements for the scheduling problems to that of the different planning periods for each product. In addition to constraints (1a-1e), the model also includes detailed scheduling constraints (1f-1r) for each production site (p ∈ PS) and for each planning period (t ∈ T). These scheduling level constraints are allocation constraints (1f), production capacity constraints (1g), storage capacity constraints (1h), material balance constraints (1i) and (1j), and sequence constraints (1k-1r). Equations (1a-1r) comprise the complete multisite batch production facilities and multisite distribution centers considering planning and scheduling decisions.

2.4. Solution Method

The full-scale integrated model gives rise to a large scale optimization problem which requires the use of decomposition methods to be solved effectively. The appropriate mathematical decomposition approach is decided by analyzing the constraints matrix of the full-scale model. If the planning decision variables ($Inv_s^{p,t}, P_s^{p,t}, D_s^{p,m,t}, U_s^{m,t}$) are denoted as $X^{t,p}$ and scheduling decision variables as $Y^{t,p}$, then the structure of the integrated model can be illustrated as shown in a constraints matrix (Figure 2.2). As it is seen in the Figure 2.2, the matrix has a block angular structure and these blocks are linked through the planning decisions variables, inventory and production targets for each production facilities ($Inv_s^{p,t}, P_s^{p,t}$). These complicating variables can be handled using augmented Lagrangian relaxation method described in the next section to obtain a decomposable structure.

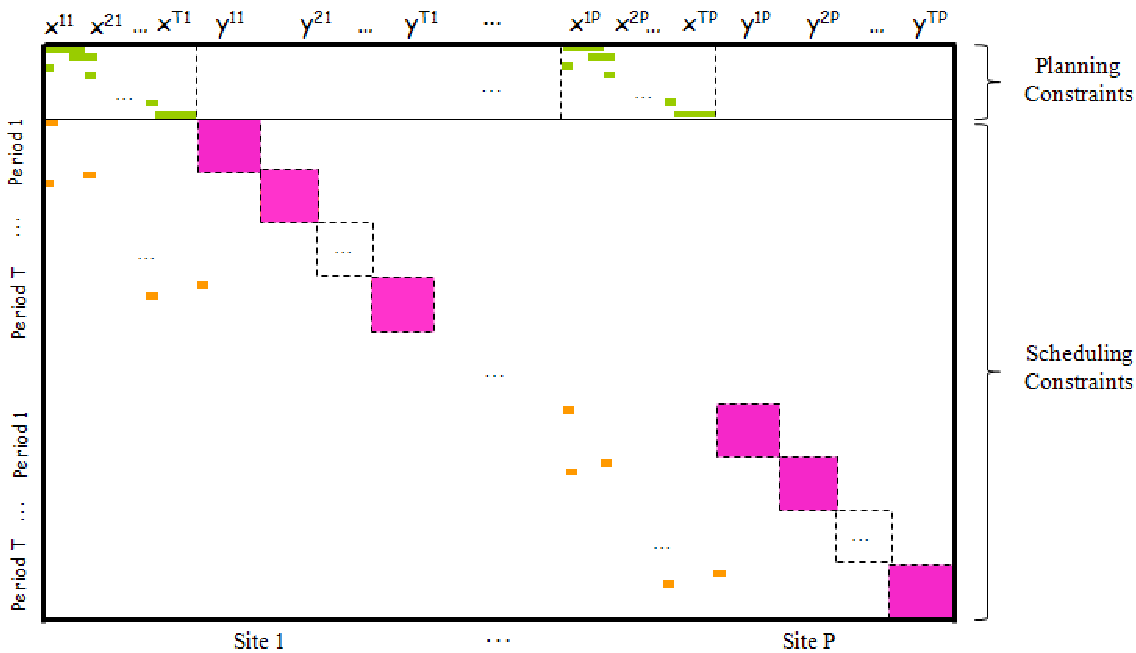

Figure 2.2 Constraints matrix structure of an integrated multisite model

2.4.1. Augmented Lagrangian Decomposition

In order to obtain a decomposable structure, the complicating variables need to be transformed into complicating constraints and then, the model can be relaxed by eliminating complicating constraints from the total constraints set. The first step in obtaining the relaxation problem is to duplicate the planning inventory and production targets variables, using different variables in planning and scheduling problems and incorporate the coupling constraints (2f-2g) into the full-scale model. The production and inventory scheduling targets constraints are rewritten as (2q-2r).

This transforms the constraints matrix (Figure 2.2) into the matrix with complicating constraints shown in Figure 2.3.

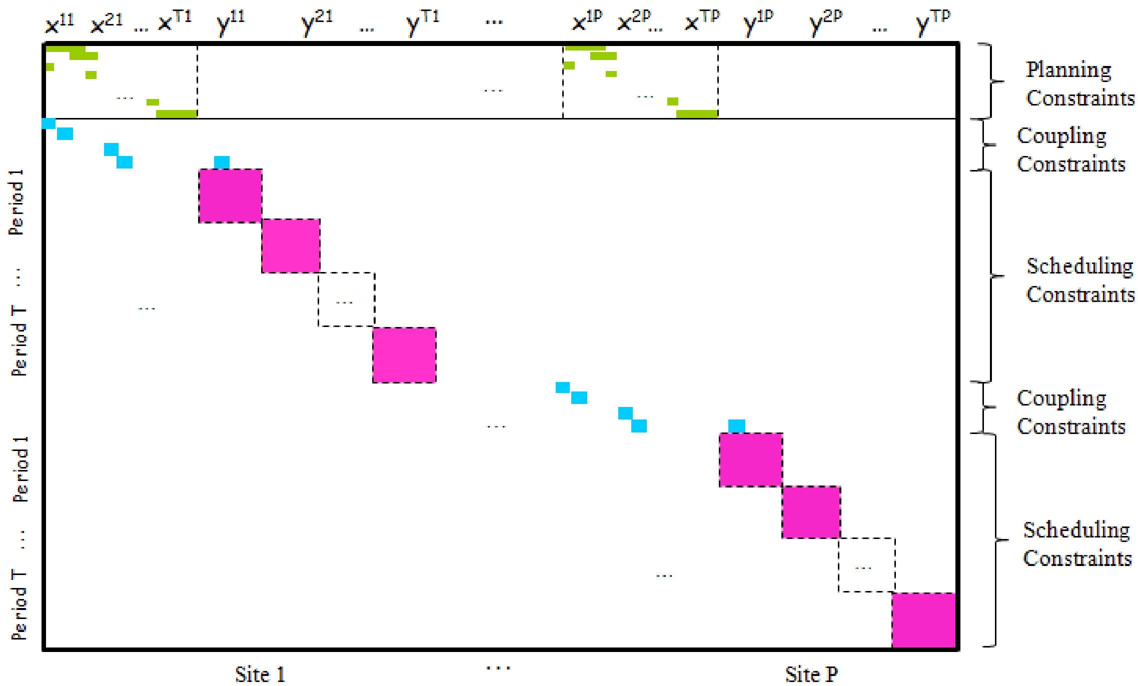

Figure 2.3 Constraints matrix structure of a reformulated model

$$\min \sum_t \sum_p \sum_{s \in S_f^p} h_s^p Inv_s^{p,t} + \sum_t \sum_m \sum_{s \in S_f^m} u_s^m U_s^{m,t} + \sum_t \sum_m \sum_p \sum_{s \in S_f^p \cap S_f^m} d_s^{p,m} D_s^{p,m,t}$$
$$+ \sum_t \sum_p \sum_{i \in I^p} \sum_{j \in J_i^p} \sum_n \left(FixCost_i^p wv_{i,j,n}^{p,t} + VarCost_i^p b_{i,j,n}^{p,t} \right) \qquad (2a)$$

$$s.t. \quad Inv_s^{p,t} = Inv_s^{p,t\text{-}1} + P_s^{p,t} - \sum_m D_s^{p,m,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \qquad (2b)$$

$$U_s^{m,t} = U_s^{m,t\text{-}1} + Dem_s^{m,t} - \sum_{p \in P_s} D_s^{p,m,t}, \quad \forall s \in S_f^m, m \in M, t \in T \qquad (2c)$$

$$st_{s,n=N}^{p,t} - stin_s^{p,t} = PP_s^{p,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \qquad (2q)$$

$$stin_s^{p,t} = II_s^{p,t\text{-}1}, \quad \forall s \in S_f^p, p \in PS, t \in T \qquad (2r)$$

$$II_s^{p,t} = Inv_s^{p,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \qquad (2f)$$

$$PP_s^{p,t} = P_s^{p,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{2g}$$

$$y^{p,t} \in Y, \quad \forall p \in PS, t \in T \tag{2h}$$

The complicating constraints (2f-2g) link the planning level constraints with the scheduling level constraints. The constraints (2h) express a compact representation of the scheduling level constraints (1f-1r) for each production site and planning period.

The reformulated model (2) is still not decomposable since the planning and scheduling problems are interconnected via coupling constraints thus the Augmented Lagrangian relaxation method is applied by dualizing the complicating constraints in equations (2f-2g), which involves removing them from the reformulated model constraints set and adding them to the objective function multiplied by the Lagrange multipliers ($\lambda_s^{p,t}$, $\mu_s^{p,t}$) and quadratic penalty parameters (σ) as shown in equation (3a). Constraints (3a-3h) correspond to the augmented Lagrangian relaxation problem.

$$\begin{aligned} f(\lambda,\mu,\sigma) = \min & \sum_t \sum_p \sum_{s \in S_f^p} h_s^p Inv_s^{p,t} + \sum_t \sum_m \sum_{s \in S_f^m} u_s^m U_s^{m,t} + \sum_t \sum_m \sum_p \sum_{s \in S_f^p \cap S_f^m} d_s^{p,m} D_s^{p,m,t} \\ & + \sum_t \sum_p \sum_{i \in I^p} \sum_{j \in J_i^p} \sum_n \left(FixCost_i^p wv_{i,j,n}^{p,t} + VarCost_i^p b_{i,j,n}^{p,t} \right) + \sum_t \sum_{s \in S_f^p} \lambda_s^{p,t} \left(P_s^{p,t} - PP_s^{p,t} \right) \\ & + \sum_t \sum_p \sum_{s \in S_f^p} \mu_s^{p,t} \left(Inv_s^{p,t} - II_s^{p,t} \right) + \sum_t \sum_p \sum_{s \in S_f^p} \sigma \left\{ \left(P_s^{p,t} - PP_s^{p,t} \right)^2 + \left(Inv_s^{p,t} - II_s^{p,t} \right)^2 \right\} \end{aligned} \tag{3a}$$

$$s.t. \quad Inv_s^{p,t} = Inv_s^{p,t-1} + P_s^{p,t} - \sum_m D_s^{p,m,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{3b}$$

$$U_s^{m,t} = U_s^{m,t-1} + Dem_s^{m,t} - \sum_{p \in P_s} D_s^{p,m,t}, \quad \forall s \in S_f^m, m \in M, t \in T \tag{3c}$$

$$st_{s,n=N}^{p,t} - stin_s^{p,t} = PP_s^{p,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{3q}$$

$$stin_s^{p,t} = II_s^{p,t-1}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{3r}$$

$$y^{p,t} \in Y, \quad \forall p \in PS, t \in T \tag{3h}$$

To improve the convergence to the feasible solution and to avoid duality gap that may result with just the Lagrangian terms ($\lambda_s^{p,t}\left(P_s^{p,t} - PP_s^{p,t}\right)$ and $\mu_s^{p,t}\left(Inv_s^{p,t} - II_s^{p,t}\right)$), the quadratic penalty term ($\sigma\left\{\left(P_s^{p,t} - PP_s^{p,t}\right)^2 + \left(Inv_s^{p,t} - II_s^{p,t}\right)^2\right\}$) is applied to the relaxation formulation as given in (3a). However, the quadratic penalty term in the objective function of the relaxation problem has non-separable bilinear terms $P_s^{p,t} \cdot PP_s^{p,t}$ and $Inv_s^{p,t} \cdot II_s^{p,t}$. To resolve the non-separabilility issue, the diagonal quadratic approximation (DQA) method is applied to linearize the cross-product terms around the tentative solution $\overline{P}_s^{p,t}, \overline{PP}_s^{p,t}, \overline{Inv}_s^{p,t}, \overline{II}_s^{p,t}$ as shown in the following equations ((Y. Li et al., 2008)).

$$\left(Inv_s^{p,t} - II_s^{p,t}\right)^2 \approx \left(Inv_s^{p,t} - \overline{II}_s^{p,t}\right)^2 + \left(\overline{Inv}_s^{p,t} - II_s^{p,t}\right)^2 - \left(\overline{Inv}_s^{p,t} - \overline{II}_s^{p,t}\right)^2$$

$$\left(P_s^{p,t} - PP_s^{p,t}\right)^2 \approx \left(P_s^{p,t} - \overline{PP}_s^{p,t}\right)^2 + \left(\overline{P}_s^{p,t} - PP_s^{p,t}\right)^2 - \left(\overline{P}_s^{p,t} - \overline{PP}_s^{p,t}\right)^2$$

The objective function (3a) can be thus be rewritten in decomposable form given by equation (4a').

$$f(\lambda, \mu, \sigma) = f_{pl} + \sum_{t \in T} \sum_{p \in PS} f_{sc}^{p,t} \tag{4a'}$$

where, the f_{pl} represents the objective function of the planning problem (4a, 4b, 4c).

$$\begin{aligned} f_{pl}\left(\lambda, \mu, \sigma\right) = \min_{P, Inv, D, U} & \sum_t \sum_p \sum_{s \in S_f^p} h_s^p Inv_s^{p,t} + \sum_t \sum_m \sum_{s \in S_f^m} u_s^m U_s^{m,t} + \sum_t \sum_m \sum_p \sum_{s \in S_f^p \cap S_f^m} d_s^{p,m} D_s^{p,m,t} \\ & + \sum_t \sum_{s \in S_f^p} \lambda_s^{p,t} P_s^{p,t} + \sum_t \sum_p \sum_{s \in S_f^p} \mu_s^{p,t} Inv_s^{p,t} \\ & + \sum_t \sum_p \sum_{s \in S_f^p} \sigma\left\{\left(P_s^{p,t} - \overline{PP}_s^{p,t}\right)^2 + \left(Inv_s^{p,t} - \overline{II}_s^{p,t}\right)^2 - \left(\overline{P}_s^{p,t} - \overline{PP}_s^{p,t}\right)^2 - \left(\overline{Inv}_s^{p,t} - \overline{II}_s^{p,t}\right)^2\right\} \end{aligned} \tag{4a}$$

$$s.t.\quad Inv_s^{p,t} = Inv_s^{p,t\text{-}1} + P_s^{p,t} - \sum_m D_s^{p,m,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{4b}$$

$$U_s^{m,t} = U_s^{m,t\text{-}1} + Dem_s^{m,t} - \sum_{p \in P_s} D_s^{p,m,t}, \quad \forall s \in S_f^m, m \in M, t \in T \tag{4c}$$

where $f_{sc}^{p,t}$ represents the objective function of the scheduling sub-problems (5a, 5b-5d). The scheduling sub-problem is defined at each production site and for each planning period.

$$\begin{aligned} f_{sc}^{p,t}(\lambda,\mu,\sigma) = \min_{PP^{p,t}, II^{p,t}, y^{p,t}} & \sum_{i \in I^p} \sum_{j \in J_i^p} \sum_n \left(FixCost_i^p wv_{i,j,n}^{p,t} + VarCost_i^p b_{i,j,n}^{p,t} \right) - \sum_{s \in S_f^p} \lambda_s^{p,t} PP_s^{p,t} \\ & - \sum_{s \in S_f^p} \mu_s^{p,t} II_s^{p,t} + \sum_{s \in S_f^p} \sigma \left\{ \left(\overline{P}_s^{p,t} - PP_s^{p,t} \right)^2 + \left(\overline{Inv}_s^{p,t} - II_s^{p,t} \right)^2 \right\} \end{aligned} \tag{5a}$$

$$s.t.\quad st_{s,n=N}^{p,t} - stin_s^{p,t} = PP_s^{p,t}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{5b}$$

$$stin_s^{p,t} = II_s^{p,t\text{-}1}, \quad \forall s \in S_f^p, p \in PS, t \in T \tag{5c}$$

$$y^{p,t} \in Y, \quad \forall p \in PS, t \in T \tag{5d}$$

These quadratic problems are solved using a general augmented Lagrangian optimization and diagonal quadratic approximation (ALO-DQA) algorithm which is outlined in Figure 2.4.

The ALO-DQA algorithm can provide an optimal solution if the objective functions and constraints function are convex, feasible region is bounded and closed, and the step size is sufficiently small. The algorithm has the following parameters: the initial Lagrange multipliers ($\lambda_s^{p,t,0}, \mu_s^{p,t,0}$) which are chosen to be zero, the initial penalty parameter ($\sigma^0 > 1$), and the iteration counter k which is set to 1. The convergence tolerance ($\varepsilon > 0$) for the coupling constraints is 1 and the parameters $\beta \in (0,1)$ (e.g., 0.4) and $\alpha > 1$.

The augmented Lagrangian multipliers are updated at every iteration as shown in Figure 2.4 while the quadratic penalty parameters are updated only if the improvement of the current iteration is not large enough. The ALO-DQA method alternates between solving an optimization planning problem (4) and solving optimization scheduling sub-problems (5). The solution of each problem is used to linearize the non-separable terms and the algorithm terminates when the consistency constraints (g) have met the pre-defined tolerance or when the given iteration limit is reached. In the next section, three numerical examples are solved using the ALO-DQA method.

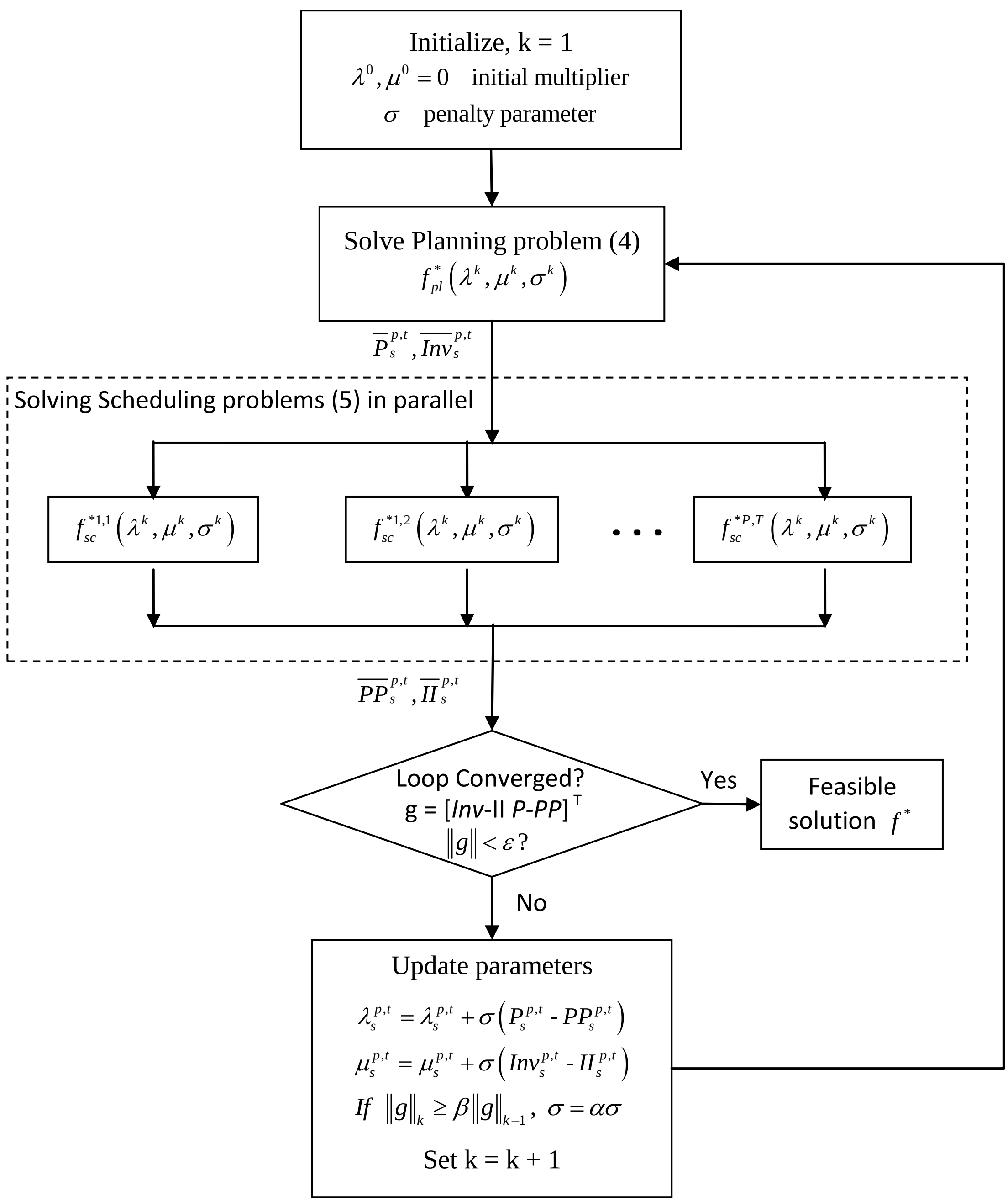

Figure 2.4 Augmented Lagrangian-DQA decomposition algorithm

2.5. Numerical Examples

The proposed multiproduct and multisite production facility model is applied to the two examples of supply chain with a planning horizon of 3 months (90 days) and scheduling horizon of 1 production shift (8 hours). The full-scale integrated planning and scheduling problem corresponds to a mixed integer linear programming (MILP) problem while in the ALO-DQA method, the planning problem is quadratic programming (QP) problem and scheduling sub-problems are mixed integer quadratic programming (MIQP) problem. The multisite models were implemented using GAMS 23.6 (2010) and solved using CPLEX 12.2 on 2.53 GHz Precision T7500 Tower Workstation with 6 GB RAM. The scheduling sub-problems (MIQP) in the ALO-DQA method are solved in parallel for each planning period and each production site thus improving the efficiency of the algorithm. In all the examples studied in our work, limited storage capacity for final products and intermediate materials is considered.

Example 1: A small example that has 3 production sites serving 3 markets is studied. Each production site contains a multiproduct, multitask batch process plant that produces two products, P1 and P2. (Kondili et al., 1993) The state and task representation (STN) of the production plant is given in Figure 2.5 and the data for the example are given in Appendix Chapter 2. The process parameters, production and inventory costs, and shipping and backorder costs are given in Table A2-1, Table A2-2, and Table A2-3, respectively. Continuous time scheduling problem is solved using 6 event points and 8 hour time horizon. The daily demand data for the example 1 is given in Figure A2-1 for planning horizon of 90 days.

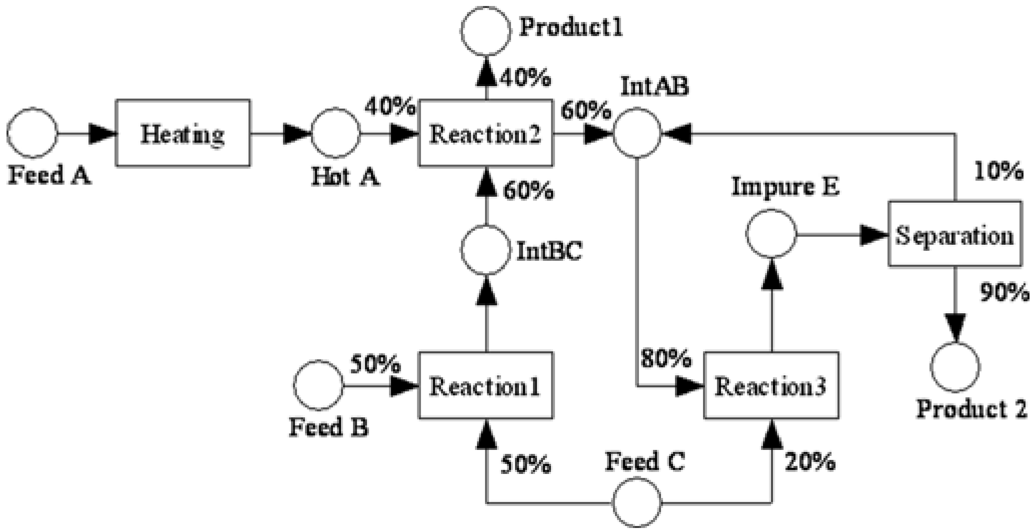

Figure 2.5 Production facility state and task network (STN) representation (Example 1)

Table 2.1 Model statistics of full-space integrated problem

Period (T)	Example 1			Example 2			Example 3		
	Binary var.	Cont. var.	Const.	Binary var.	Cont. var.	Const.	Binary var.	Cont. var.	Const.
5	720	4006	10465	720	5011	10654	1440	8851	21178
10	1440	8011	20935	1440	10021	21319	2880	17701	42373
15	2160	12016	31405	2160	15031	31984	4320	26551	63568
30	4320	24031	62815	4320	30061	63979	8640	53101	127153
45	6480	36046	94225	6480	45091	95974	12960	79651	190738
60	8640	48061	125635	8640	60121	127969	17280	106201	254323
90	12960	72091	188455	12960	90181	191959	25920	159301	381493

The full-scale model statistics for example 1 is shown in Table 2.1 and results are given in Table 2.2 for time periods 5 to 90. As the time periods increases, the problem becomes difficult to solve to optimality as observed by the optimality gap (%) in Table 2.2 for the full-space model. The performance and quality of the full-scale model to that of the ALO-DQA is compared in Table

2.2. From the Figure 2.6, it can be observed that the augmented Lagrangian algorithm converges to a feasible solution of the original problem as the norm value of the coupling constraints (||g||) converges to zero. The quality of the feasible solution (f*) obtained using the ALO-DQA method may be inferior to the full-scale model since the ALO-DQA strategy solves an approximation version of the relaxation problem. The key information that the integrated model solution provides is production, shipping, sales, inventory, and backorder profiles. The profiles for example 1 obtained using the ALO-DQA method are shown in Figure 2.7 to Figure 2.10 for 30 planning periods. The production profiles for production sites (S1, S2, and S3) are shown in Figure 2.7. The transportation profile of products (P1 and P2) from production site (S1, S2, and S3) to market place (M1, M2, and M3) is given in Figure 2.8, Figure 2.9, and Figure 2.10. Note that as shown in Figure 2.8, Figure 2.9, and Figure 2.10, the production sites 1, 2, and 3 mainly satisfy the demand of markets 1, 2, and 3, respectively. These transportation profiles are expected based on the shipping cost. The total sale of products (P1 and P2) at markets (M1, M2, and M3) is given in Figure 2.11. The variable inventory holding cost is highest at production site 2 and lowest at site 3 and the model solution gives an inventory profiles (Figure 2.12) that has highest holdup at site 3 and lowest holdup at site 2. As expected, the advantage of the proposed decomposition approach is shown for bigger problems. So for larger number of time periods (T=90 periods), better solutions were obtained using the ALO-DQA method than by the full-scale model.

Table 2.2 Computational results for example 1

T	**Full-space model**			**ALO-DQA method** $\sigma^0 = 1.0, \alpha = 2.0$				
	CPU sec	f*	Gap (%)	Iter. k	CPU sec	f*	$\lambda g + \sigma\|\|g\|\|^2$	$\|\|g\|\|$
5	3600	58073	0.63	10	34.68	59500	0.62	0.22

10	3600	119227	1.91	9	53.53	122903	34.80	0.66
15	3600	194567	2.07	12	101.97	199983	-157.42	0.89
30	3600	390528	2.27	12	183.21	403208	-17.96	0.85
45	3600	592649	2.44	12	292.25	608059	-166.23	0.76
60	7200	791399	3.28	13	436.03	803867	-128.27	0.59
90	7200	1225376	6.93	15	760.50	1203194	-324.10	0.92

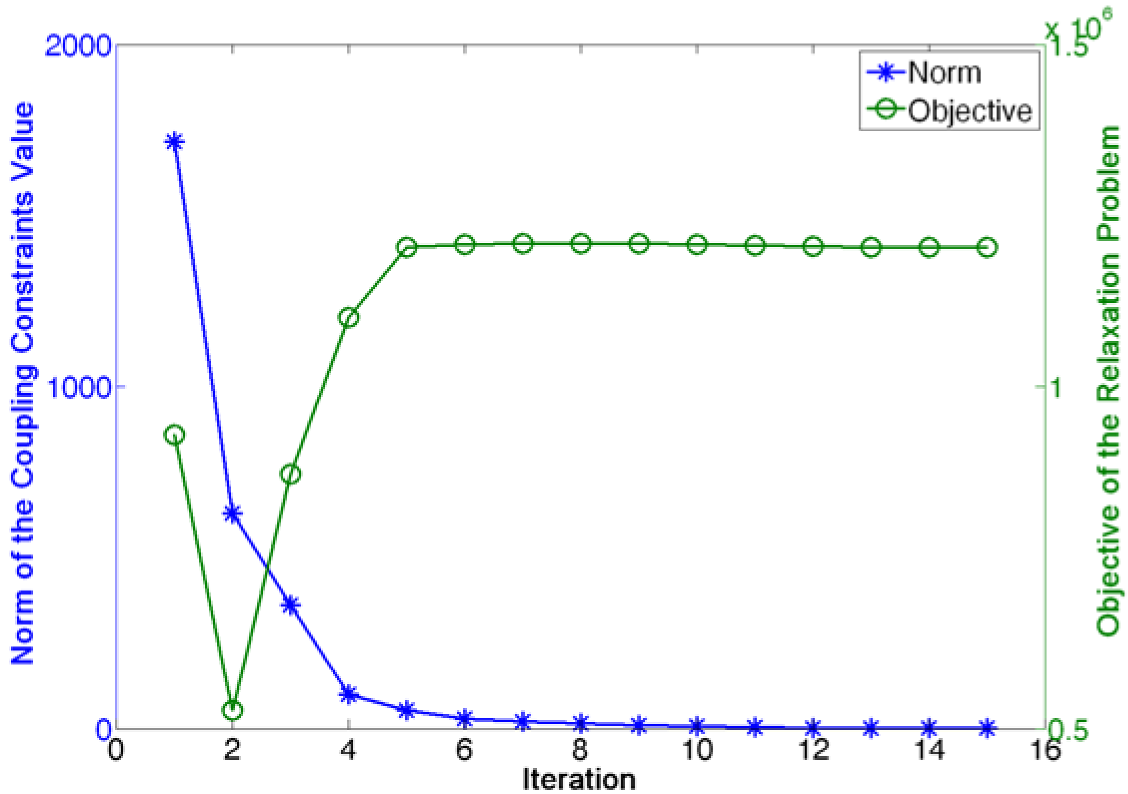

Figure 2.6 Solution procedure of the ALO-DQA method (Example 1 and 90 planning periods)

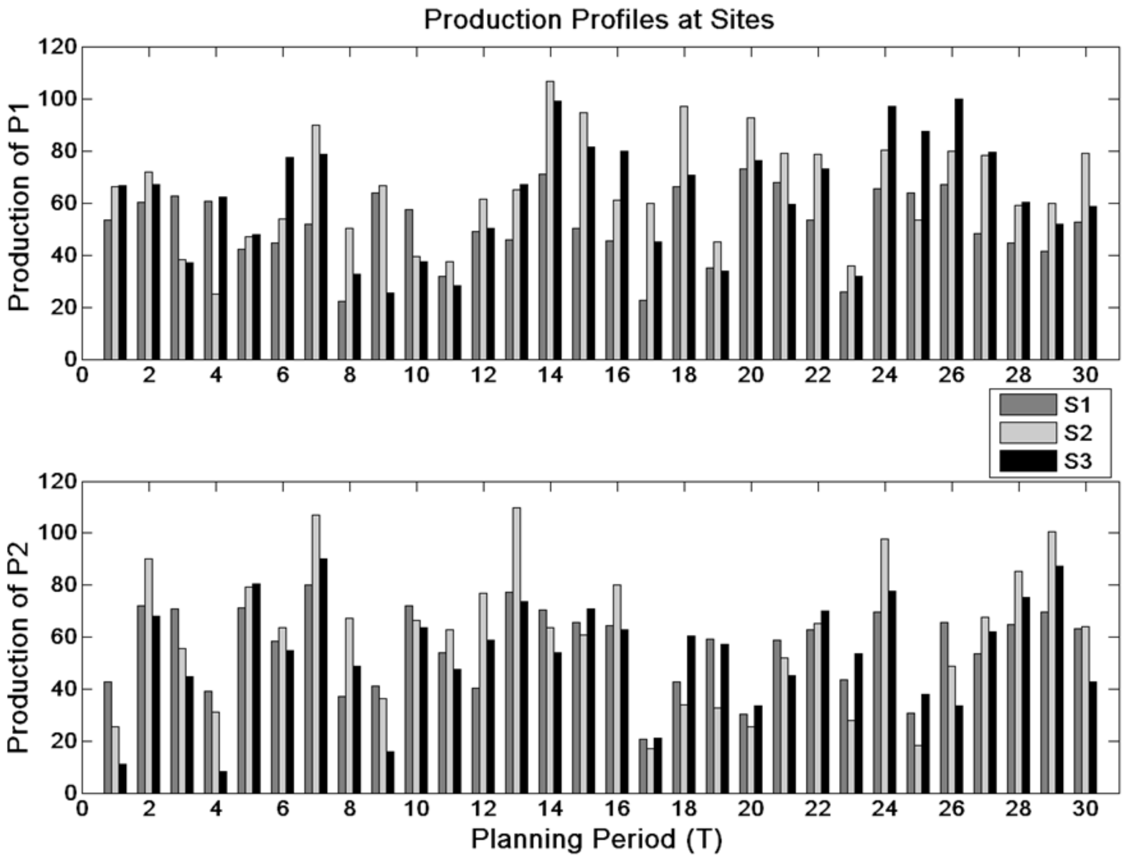

Figure 2.7 Production profiles of products (P1 and P2) at production sites (S1, S2, and S3) obtained using the ALO-DQA method (Example 1 and 30 planning periods)

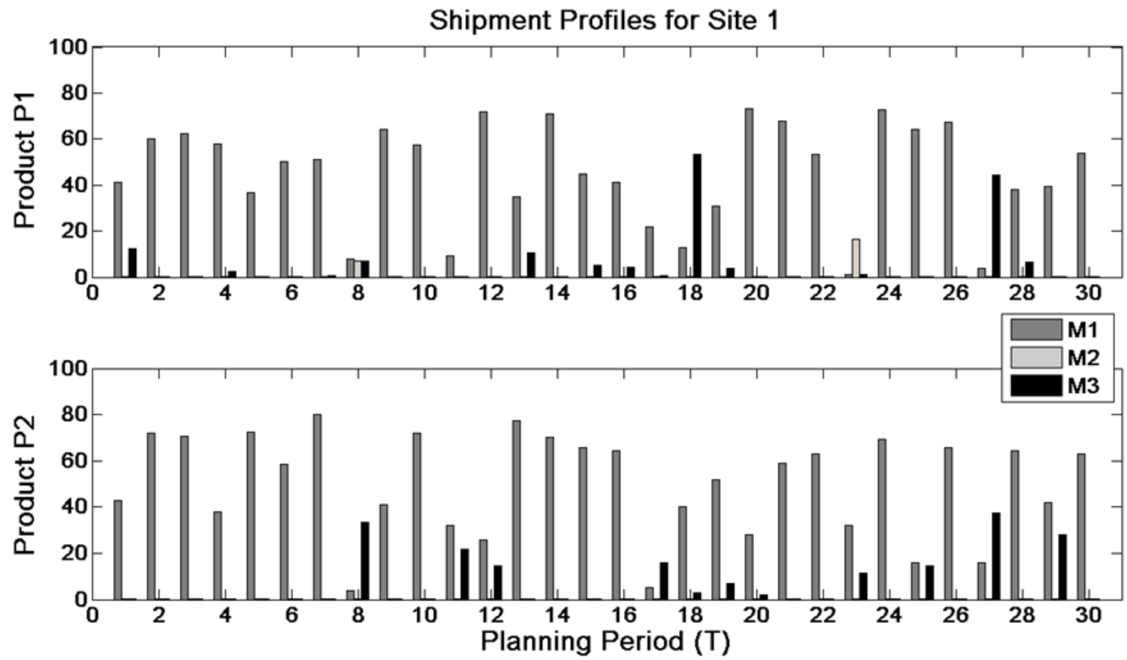

Figure 2.8 Shipment profiles of products (P1 and P2) for production site 1 to markets (M1, M2, and M3) obtained using the ALO-DQA method (Example 1 and 30 planning periods)

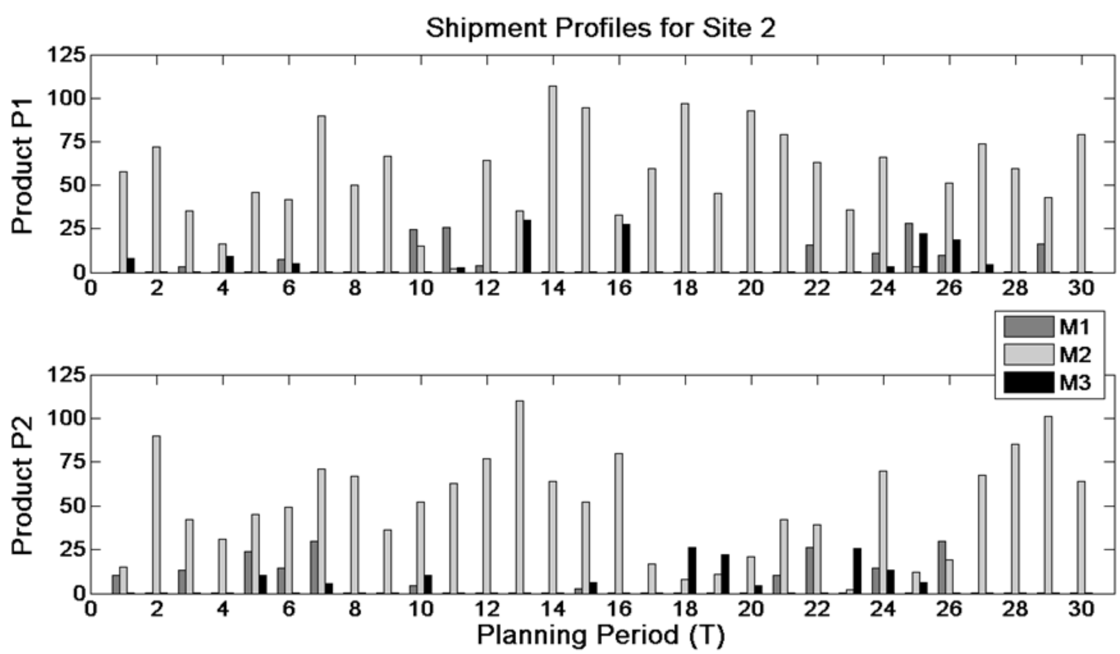

Figure 2.9 Shipment profiles of products (P1 and P2) for production site 2 to markets (M1, M2, and M3) obtained using the ALO-DQA method (Example 1 and 30 planning periods)

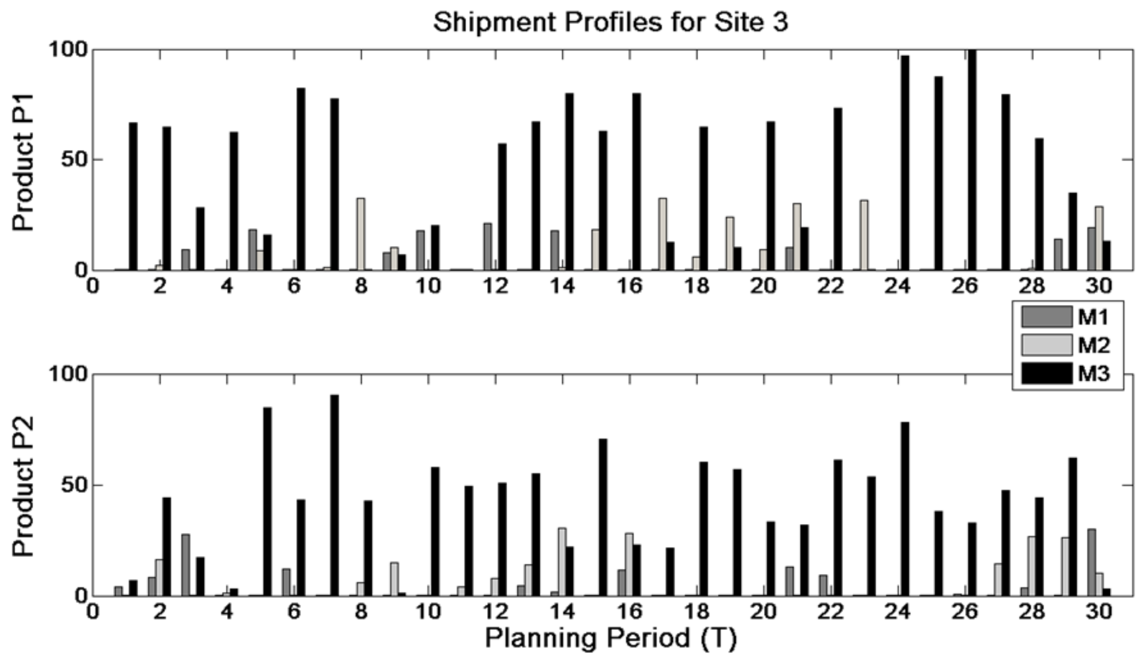

Figure 2.10 Shipment profiles of products (P1 and P2) for production site 3 to markets (M1, M2, and M3) obtained using the ALO-DQA method (Example 1 and 30 planning periods)

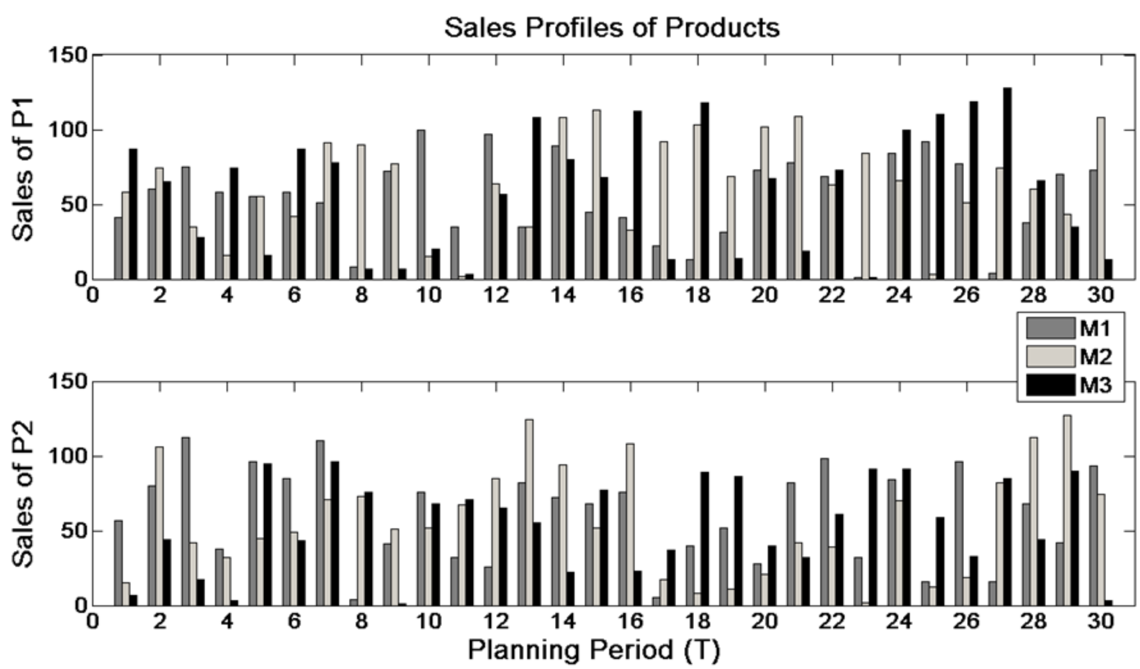

Figure 2.11 Sales profiles of products (P1 and P2) for markets (M1, M2, and M3) obtained using the ALO-DQA method (Example 1 and 30 planning periods)

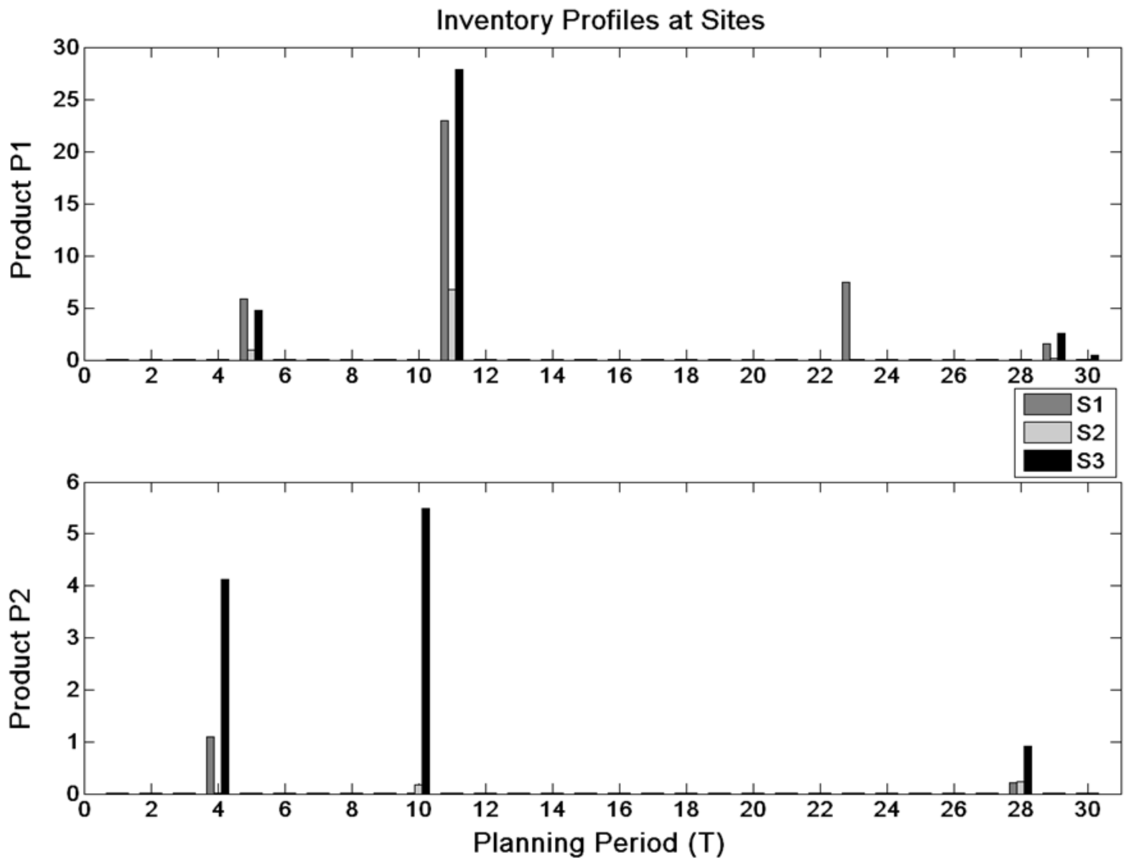

Figure 2.12 Inventory profiles of products (P1 and P2) at production sites (S1, S2, and S3) obtained using the ALO-DQA method (Example 1 and 30 planning periods)

Example 2. In example 2, we consider a network of 3 production sites producing 4 different products (P3-P6) and serving 3 global markets. All 3 production sites have batch facilities whose STN network is shown in Figure 2.13. (Kondili, 1987) This batch facility produces 4 products (P3, P4, P5, and P6) through 8 tasks from three feeds and there are 6 intermediates state in the system. The full-scale model statistics for this example are shown in Table 2.1 and results are shown in Table 2.3. The full-scale problem is much easier to solve compared with example 1, even though this example is bigger than example 1. Thus, the production recipe, and the parameters relating to production, capacity, demand, and costs have significant effect on the performance of the full-scale model. To further improve the integrated model performance, we applied the ALO-DQA method and the results are shown in Table 2.3. The solution convergence profile for planning period 90 is shown in Figure 2.14. The performance of the ALO-DQA method depends on the choice of the initial values of Lagrange multipliers, penalty parameters, and other algorithm parameters. By selecting appropriate values of these parameters, we can improve on the quality of the feasible solution obtained by the ALO-DQA. The results with set of parameters $\lambda^0 = 0, \mu^0 = 0, \sigma^0 = 0.20, \alpha = 1.2, \beta = 0.4$ are shown in the Table 2.3. Significant CPU time savings is reported when the ALO-DQA method is used compared to the integrated full-scale model.

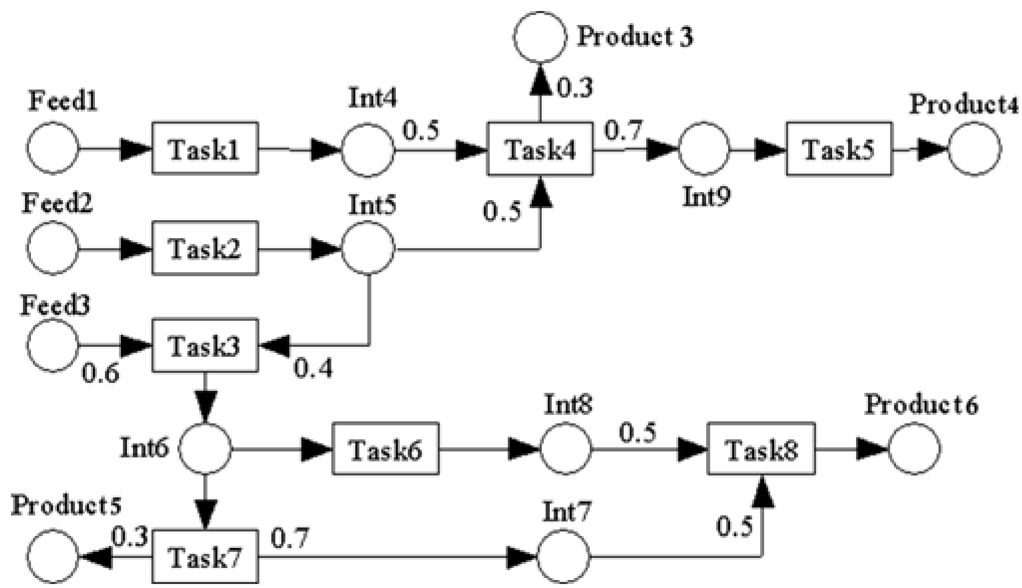

Figure 2.13 Production facility state and task network (STN) representation (Example 2)

Table 2.3 Computational results for example 2

T	**Full-space model**			**ALO-DQA method** $\sigma^0 = 0.2, \alpha = 1.2$				
	CPU sec	f*	Gap (%)	Iter k	CPU sec	f*	$\lambda g + \sigma\|g\|^2$	$\|g\|$

5	52.15	249097	0.00	36	49.37	250052	99.31	0.56
10	3600	502983	0.07	36	66.85	504687	14.18	0.65
15	3600	756527	0.25	40	112.49	762006	-211.75	0.89
30	3600	1381017	0.50	40	209.29	1390602	-216.91	0.95
45	3600	1964466	1.05	40	262.49	1968542	-234.33	0.98
60	3600	2594945	0.75	41	499.90	2608491	-1.59	0.70
90	3600	4009291	1.04	41	709.48	4024795	24.81	0.75

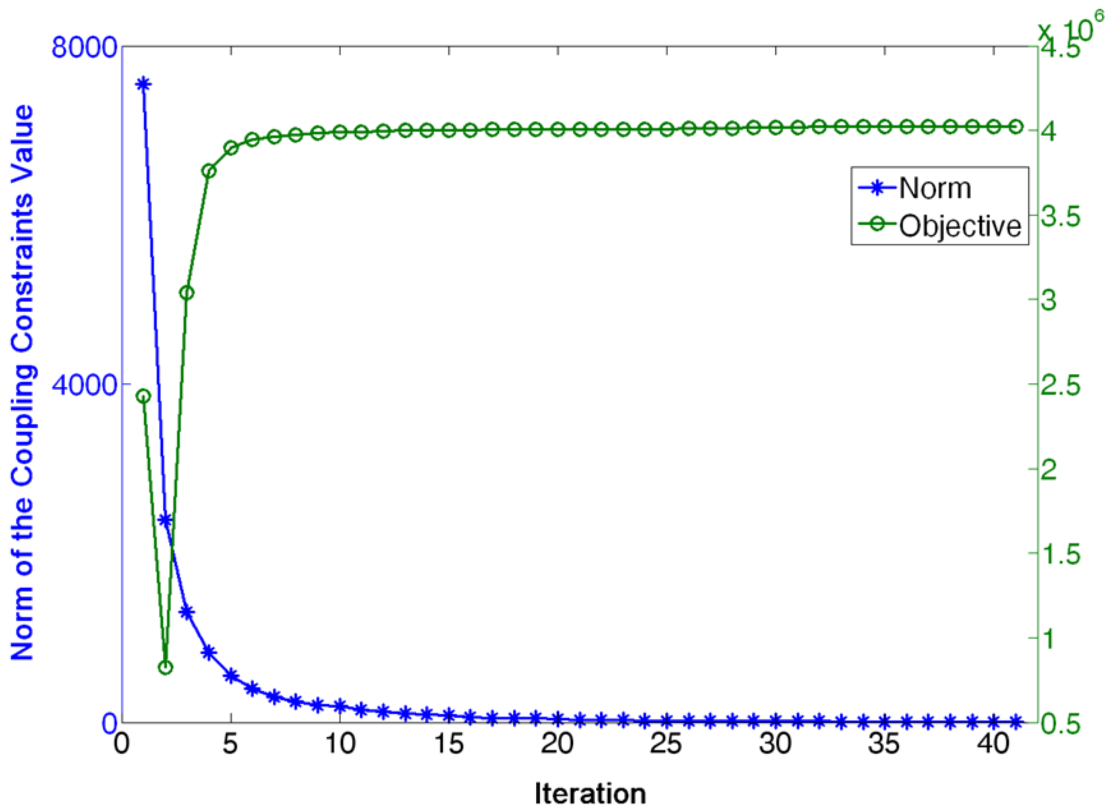

Figure 2.14 Solution procedure of the ALO-DQA method (Example 2, T = 90 periods)

Example 3. In example 3, we consider a network of 6 production sites producing 6 different products (P1-P6) and serving 9 global markets. Of the 6 production sites, 3 batch production facilities have the network shown in Figure 2.5 and produce 2 products (P1 and P2) and 3 production sites have the batch facilities whose STN network is shown in Figure 2.13 and produce 4 products (P3, P4, P5, and P6) through multipurpose units.

Table 2.4 Computational results for example 3

T	**Full-space model**			**ALO-DQA method** $\sigma^0 = 1.0, \alpha = 2.0$				
	CPU sec	f*	Gap (%)	Iter k	CPU sec	f*	$\lambda g + \sigma\|\|g\|\|^2$	$\|\|g\|\|$
5	3600	141129	4.11	11	71.11	151551	24.28	0.99
10	3600	276230	4.40	13	144.97	295756	609.56	0.88
15	3600	408436	4.97	12	192.53	435078	200.49	0.58
30	3600	784535	6.05	13	378.51	822924	-31.96	0.91
45	3600	1259638	12.97	16	749.96	1233187	-905.25	0.80
60	3600	8123074	82.11	16	984.92	1629968	-718.61	0.76
90	3600	8445397	74.11	15	1263.73	2437213	-1271.17	0.85

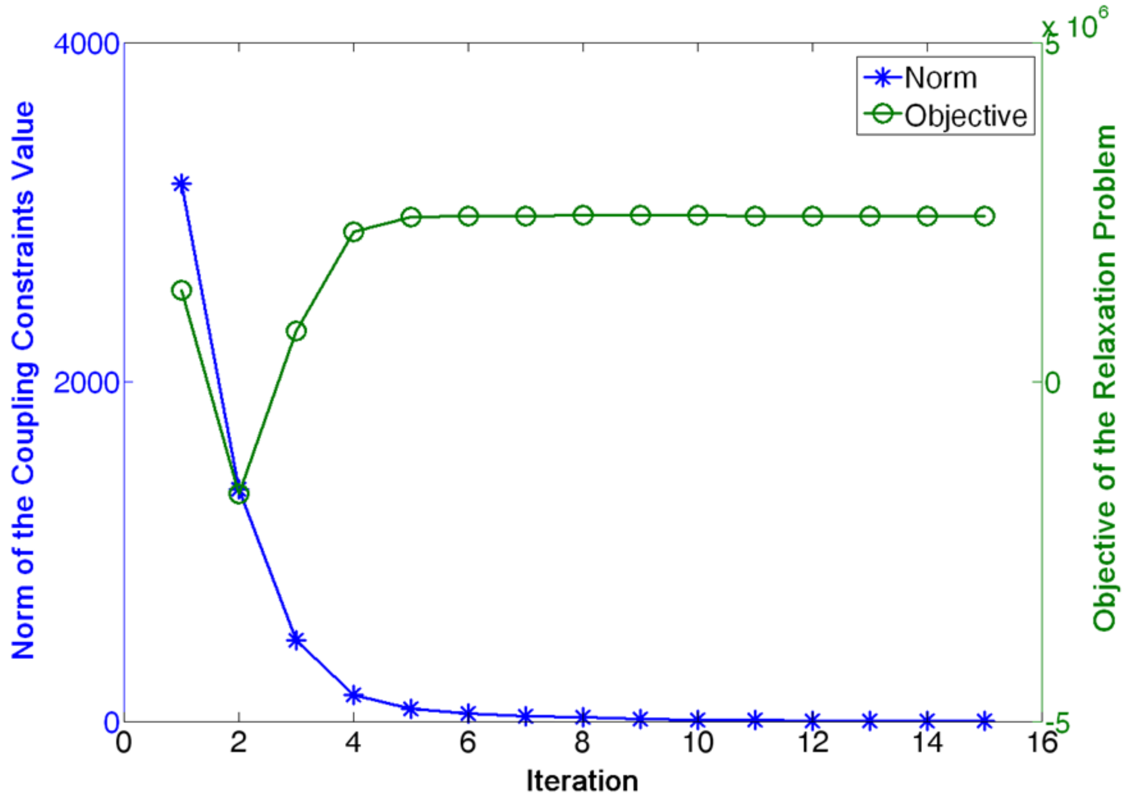

Figure 2.15 Solution procedure of the ALO-DQA method (Example 3, T = 90 periods)

The model statistics of example 3 are shown in Table 2.1. As expected, the complexity of the integrated full-scale model increases as the planning horizon increases and this is reflected in the solution time and relative gap (%) given in Table 2.4. The progress of the solution procedure of the ALO-DQA method for 90 time periods is shown in Figure 2.15. Table 2.4 shows the solutions of the integrated full-scale problem and the ALO-DQA decomposition method. Similar to results of examples 1 and 2, significant computational savings are observed when decomposition is applied compared to the full-scale model. Furthermore, for example 3, the ALO-DQA method is able to provide a better solution than the one reported by the full-scale model for planning periods of T=45, 60, and 90.

2.6. Summary

This work addresses the problem of integrated planning and scheduling for multisite, multiproduct and multipurpose batch plants using the augmented Lagrangian method. The integrated multisite model is proposed by extending single site formulation of(Z. Li & Ierapetritou, 2010a). The shipping costs from the production sites to distribution markets are taken into account explicitly in the integrated problem. Given the fixed demand forecast the model optimizes the production, inventory, transportation, and backorder costs. Temporal decomposition scheme was developed to address the large scale model resulting from multiperiod planning and scheduling problem. The augmented Lagrangian relaxation with diagonal approximation method allowed solution of the scheduling optimization problems into parallel. Three example problems solved to illustrate the advantages of applying the augmented Lagrangian decomposition scheme. With the proposed decomposition method, faster solution times were realized.

Nomenclature

Indices

i	task
j	units
m	distribution market
n	event point
p	production site
s	material state
t	planning period

Sets

I	tasks

I^p	task that can be performed at site *p*
I_j^p	tasks that can be performed in unit *j* at site *p*
J	units
J^p	units that are located at site *p*
J_i^p	units that can perform task *i* at site *p*
M	distribution markets
N	event points
P_s	sites that can produced final product *s*
PS	production sites
S	material states
S_f^m	products that can be sold at market *m*
S_f^p	products that can be produced at site *p*
T	planning periods

Parameters

$d_s^{p,m}$	unit transport cost of material *s* from site *p* to market *m*
$Dem_s^{m,t}$	demand of product *s* at market *m* for period *t*
$FixCost_i^p$	fixed production cost of task *i* at site *p*
h_s^p	holding cost of product *s* at production site *p*
$stcap_s^p$	available maximum storage capacity for state *s* at site *p*
u_s^m	backorder cost of product *s* at distribution market *m*
$V_{i,j,p}^{\min}, V_{i,j,p}^{\max}$	minimum and maximum capacity of unit *j* when processing task *i* at site *p*
$VarCost_i^p$	unit variable cost of task *i* at site *p*

$\alpha_{i,j}^{p}$, $\beta_{i,j}^{p}$ constant, variable term of processing time of task *i* in unit *j* at site *p*

$\rho_{s,i}^{c,p}$, $\rho_{s,i}^{p,p}$ proportion of state *s* consumed, produced by task *i* respectively at site *p*

Variables

$b_{i,j,n}^{p,t}$ amount of material processed by task *i* in unit *j* at event point *n* at site *p* during period *t*

$D_{s}^{p,m,t}$ transportation of product *s* from site *p* to market *m* at period *t*

$Inv_{s}^{p,t}$ inventory level of state *s* at the end of the planning period *t* for site *p*

$stin_{s}^{p,t}$ initial inventory for state *s* in planning period *t*

$Tf_{i,j,n}^{p,t}$ finish time of task *i* in unit *j* at event point *n* in site *p* during period *t*

$Ts_{i,j,n}^{p,t}$ start time of task *i* in unit *j* at event point *n* in site *p* during period *t*

$U_{s}^{m,t}$ backorder of product *s* at market *m* in planning period *t*

$wv_{i,j,n}^{p,t}$ binary variable, task *i* active in unit *j* at event point *n* at site *s* during period *t*

Chapter 3

3. Centralized – decentralized optimization for refinery scheduling

This chapter presents a novel decomposition strategy for solving large scale refinery scheduling problems. Instead of formulating one huge and unsolvable MILP or MINLP for centralized problem, we propose a general decomposition scheme that generates smaller sub-systems that can be solved to global optimality. The original problem is decomposed at intermediate storage tanks such that inlet and outlet stream of the tank belong to the different sub-systems. Following the decomposition, each decentralized problem is solved to optimality and the solution to the original problem is obtained by integrating the optimal schedule of each sub-systems. Different case studies of refinery scheduling are presented to illustrate the applicability and effectiveness of the decentralized strategy. The conditions under which these two types of optimization strategies (centralized and decentralized) give the same optimal result are discussed.

3.1. Introduction

Production scheduling defines which products should be produced and which products should be consumed in each time instant over a given small time horizon; hence, it defines which run-mode to use and when to perform changeovers in order to meet the market needs and satisfy the demand. Large-scale scheduling problems arise frequently in oil refineries where the main objective is to assign sequence of tasks to processing units within certain time frame such that demand of each product is satisfied before its due date. As the scale of the production problem increases, the mathematical complexity of the corresponding scheduling problem increases exponentially. Decomposition of the initial system into subsystems which are easier to be solved, is a natural way to deal with this type of optimization problems.

There are relatively few papers that have addressed planning and scheduling problems using centralized and decentralized optimization strategies providing a comparison of these two approaches. (Kelly & Zyngier, 2008)presented a procedure to find a suitable way to decompose large decision-making problems and compared different decentralized approaches using hierarchical decomposition heuristics. The focus of their work was to find globally feasible solutions to large decentralized and distributed decision-making problems when a centralized approach is not possible. (G. K. Saharidis et al., 2006; G. K. D. Saharidis, Kouikoglou, et al., 2009)studied the problem of production planning in deterministic and stochastic environments and compared centralized and decentralized optimization for an enterprise consisting of two production plants in series producing many different outputs with subcontracting options. (Chen & Chen, 2005)studied a joint replenishment arrangement with a two-echelon supply chain with one supplier and one buyer, facing a deterministic demand and selling a number of products in the marketplace. They proposed both centralized and decentralized decision policies to analyze the interplay and investigated the joint effects of two-echelon coordination and multi-product replenishment on the reduction of total costs. The cost differences between these policies show that the centralized policy significantly outperforms the decentralized policy. (Gnoni et al., 2003)present a case study from the automotive industry dealing with the lot sizing and scheduling decisions in a multi-site manufacturing system. They use a hybrid approach which combines mixed-integer linear programming model and simulation to test local and global production strategies. Their results show that local optimization strategy allows a cost reduction of about 19% compared to the reference actual annual production plan, whereas the global strategy leads to a further cost reduction of 3.5% and a better overall economic performance.(I. Harjunkoski & Grossmann, 2001) presented a decomposition scheme for solving large scheduling problems for steel production which splits the original problem into sub-systems using the special features of steel making. Their proposed approach cannot guarantee global optimality, but comparison with theoretical estimates indicates that the method produces solutions within 1-3% of the global optimum. (Bassett et al., 1996)presented resource decomposition method to reduce problem complexity by dividing the scheduling problem into subsections based on its process recipes. They showed that the overall solution time using resource decomposition is significantly lower than the time needed to solve the global problem. However, their proposed resource decomposition method did not involve any feedback mechanism to incorporate "raw material" availability between sub sections**.**

In this work, the problem of refinery scheduling optimization is addressed with centralized and decentralized decision making process. The chapter is organized as follows. Section 3.2 describes the type of problem studied in this chapter and presents a real case study provided by Honeywell Hi-Spec Solutions. Section 3.3 defines the mathematical formulation of the problem, whereas the decomposition approach is presented in section 3.4. Section 3.5 presents comparative results for centralized and decentralized optimization of the system. Finally section 3.6 draws conclusions.

3.2. Problem Definition

In general there are two decision levels in refinery process operations - the planning and the scheduling level. The planning level determines the volume of raw materials needed for the upcoming months (typically 12 months), and the type of final products and the estimated quantities to be ordered, depending on demand forecasts. After determining the yearly plan in the second level we have to determine the optimal production scheduling. The scheduling level determines the detailed schedule of each production unit and CDU for a shorter period (typically 10 days) by taking into account the operational constraints of the system under study. Once the plan is known (the quantities and the types of final products ordered as well as the arrival of raw materials), managers must schedule the production of any unit based on the objective which usually is minimization of the overall makespan or maximization of the total profit.

Refinery system considered here is composed of pipelines, a series of tanks to store the crude oil (and prepare the different mixtures), production units and tanks to store the raw materials and the intermediate and final products (see Figure 1). All the crude distillation units are considered continuous processes and it is assumed that unlimited supply of the raw material is available to system. The crude distillation unit produces different products according to the recipes. There are two types of operating scenario for the storage tanks: scenario 1 where material cannot flow out of the tank when material is flowing into the tank at any time interval, that is loading and unloading cannot happen simultaneously (due to security reasons) or scenario 2 where loading and unloading can happen simultaneously.

Figure 3.1 represents the system under study which corresponds to a really case of Honeywell Hi-Spec problem. In this system the production starts from cracking units and proceed to diesel blender unit to produce home heating oil (Red Dye diesel) and automotive diesel (Carb diesel and EPA diesel). Cracking unit, 4CU, processes Alaskan North Slope (ANS) crude oil which is stored in raw material storage tanks ANS1 and ANS2, whereas cracking unit 2 (2CU) processes San Joaquin Valley (SJV) crude oil. SJV crude oil is supplied to 2CU via pipeline.

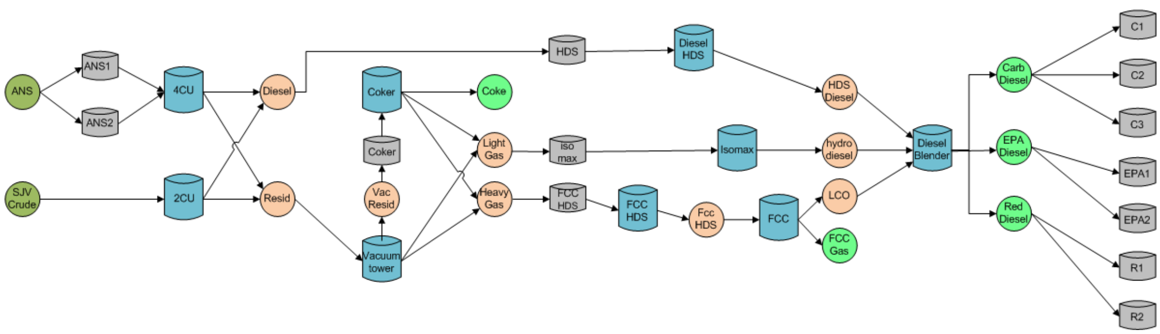

Figure 3.1 State and unit representation of Honeywell Hi-Spec problem

The products of cracking units are then processed further downstream by vacuum distillation tower unit and diesel high pressure desulfurization (HDS) unit. The coker unit converts vacuum resid into light and heavy gasoil and produces coke as residual product. The fluid catalyzed high pressure desulfurization (FCC HDS) unit, FCC, Isomax unit produce products that are needed for diesel blender unit. The FCC unit also produces by- product FCC gas. The diesel blender blends HDS diesel, hydro diesel, and light cycle oil (LCO) to produce three different final products. The diesel blender sends final products to final product storage tanks. The byproduct FCC gas and residual product Coke is not stored but supplied to the market via pipeline. The system employs four storage tanks to store intermediate products, vacuum resid, diesel, light gasoil, and heavy gasoil.

In the system studied in this chapter, the long term plan is assumed to be given and the objective is to define the optimal production scheduling. In such a case the key information available for the managers is firstly the proportion of material produced or consumed at each production units. These recipes are assumed fixed to maintain the model's linearity. The managers also know the minimum and maximum flow-rates for each production unit and the minimum and maximum inventory capacities for each storage tank. The different types of material that can be stored in each storage tank are known as well as the demand of final products at the end of time horizon. The objective is to determine the minimum total makespan of production defining the optimal values of the following variables:

1) Starting and finishing times of task taking place at each production unit;

2) Amount and type of material being produced or consumed at each time in a production unit;

3) Amount and type of material stored at each time in each tank.

There are seven groups of constraints which guarantee the operational conditions of the system. These are allocation constraints, production and storage capacity constraints, material balance constraints for units and storage tanks, demand constraints, and time sequence constraints. The mathematical formulation for the scheduling problem is presented in the following section.

3.3. Mathematical Formulation

In this section a mathematical model is presented for the scheduling of the refinery system presented in section 3.2 where the objective is to minimize the overall makespan. The developed mathematical formulation uses continuous time representation since this leads to reduced number of decision variables and constraints compared to the discrete time representation and also since due to the continuous operating mode, continuous time representation can provide more accurate results. (Ierapetritou et al., 1999; Jia et al., 2003) The mathematical formulation proposed in this chapter has the following main assumptions: 1) the change over time between different tasks at each unit is negligible; and 2) the processes are running at steady state. The proposed model presented in this section minimizes the overall makespan of the refinery production and involves allocation, capacity, storage, material balance, demand, and time sequence constraints. The integer and continuous decision variables used in the developed model give rise to a mixed integer linear programming (MILP) problem.

For the minimization of the makespan, four main operating rules have to be followed. The first one is to satisfy the constraint that at most one task can take place in one production unit at one time interval and that at most one material can be stored in one storage tank at one time interval (constraint 1). The second one is to satisfy the material balance constraints (constraints 7-10). The third one guarantees the demand satisfaction (constraint 11). Finally the forth one guarantees the correct time sequence of the tasks given that a continuous time representation is used (constraints 12-35).

Allocation Constraints:

Constraints (1) express that if a task (*i*) starts at event point (*n*), then it must be performed in one of the suitable units (*j*). It also satisfies the rule that a unit can physically perform only one task at any given time.

$$\sum_{i \in I(j)} wv(i, j, n) \le 1, \quad \forall j \in J, n \in N \tag{1}$$

Capacity Constraints:

Constraints (2) enforce the requirement that material processed by unit (*j*) performing task (*i*) at any point (*n*) is bounded by the maximum and minimum rates of production. The maximum and minimum production rates multiply by the duration of task (*i*) performed at unit (*j*) give the maximum and minimum material being processed by unit (*j*) correspondingly.

$$R^{\min}(i,j)\times\left(Tf(i,j,n)\text{-}Ts(i,j,n)\right)\le b(i,j,n)\le R_{ij}^{\max}\left(Tf(i,j,n)\text{-}Ts(i,j,n)\right),\quad \forall i\in I, j\in J(i), n\in N \tag{2}$$

Storage Constraints:

Variable $in(j,jst,n)$ is equal to 1 if there is flow of material from production unit (*j*) to storage tank (*jst*) at event point (*n*); otherwise it is equal to 0. Variable $out(jst,j,n)$ is equal to 1 if material is flowing from storage (*jst*) to unit (*j*) at event point (*n*), otherwise it is equal to 0. Equations (3) and (4) are capacity constraints for storage tank. Constraints (3) state that if there is material inflow to tank *(jst)* at interval (*n*) then total amount of material inflow to the tank should not exceed the maximum storage capacity limit. Similarly, constraints (4) state that if there is outflow from tank (*jst*) at interval (*n*) then the total amount of material flowing out of tank should not exceed the storage limit at event point (*n*).

$$infow(j,jst,n)\le V^{\max}(jst)\times in(j,jst,n),\quad \forall j\in J,\ jst\in JSTprodst(j),\ n\in N \tag{3}$$

$$outflow(jst,j,n)\le V^{\max}(jst)\times out(jst,j,n),\quad \forall j\in J,\ jst\in JSTstprod(j),\ n\in N \tag{4}$$

Constraints (5) and (6) represent the requirement that the material in tank (*jst*) should not exceed the capacity limit $V^{\max}(jst)$ of this storage tank at any event point (*n*).

$$st(jst,n-1)+\sum_{j\in Jprodst(jst)}\inf low(j,jst,n)+in\text{flow1}(jst,n)\le V^{\max}(jst),\qquad \forall jst\in Jst(s),\ n\in N \tag{5}$$

$$stin(jst)+\sum_{j\in Jprodst(jst)}in\text{flow}(j,jst,n)+in\text{flow1}(jst,n)\le V^{\max}(jst),\qquad \forall jst\in Jst(s),\ n=0 \tag{6}$$

Material Balance Constraints for Operating Unit:

Constraints (7) represent the requirement that the production of a unit should be equal to the sum of the amount of flows entering its subsequent storage tanks and reactors, and the delivery to the market.

$$\begin{aligned}\sum_{i\in I(j)}\rho^{p}(s,i)\times b(i,j,n)=\sum_{jst\in JSTprodst(j)\cap Jst(s)}in\text{flow}(j,jst,n)+\sum_{j'\in Jseq(j)\cap Junitc(s)}unitflow(s,j,j',n)\\+outflow2(s,j,n),\quad \forall s\in S,\ j\in J,\ n\in N\end{aligned} \tag{7}$$

Similarly, constraints (8) represent that the consumption at a unit is equal to the sum of the amount of streams coming from preceding storage tanks, previous units, and stream coming from supply.

$$\sum_{i\in I(j)} \rho^C(s,i)\times b(i,j,n) = inflow2(s,j,n) + \sum_{jst\in JSTstprod(j)\cap Jst(s)} outflow(j_{st},j,n) + \sum_{j'\in Jseq_j \cap Junitp_s} unitflow(s,j,j',n), \quad \forall j\in J,\ s\in S,\ n\in N \tag{8}$$

Material Balance Constraints for Storage Tank:

Similar to material balance constraints for units, the material balance constraints (9) and (10) for the storage tanks state that the inventory of a storage tank at one event point is equal to that at previous event point adjusted by the input and output stream amount.

$$St(jst,n) = St(jst,n-1) + \sum_{j\in Jprodst(jst)} inflow(j,jst,n) + inflow1(jst,n) - \sum_{j\in Jstprod(jst)} outflow(jst,j,n) - outflow1(jst,n), \quad \forall jst\in Jst,\ n\in N \tag{9}$$

$$St(jst,n) = Stin(jst) + \sum_{j\in Jprodst(jst)} inflow(j,jst,n) + inflow1(jst,n) - \sum_{j\in Jstprod(jst)} outflow(jst,j,n) - outflow1(jst,n), \quad \forall jst\in Jst,\ n\in N \tag{10}$$

Demand Constraints:

Demand for each final product *d(s)* must be satisfied in centralized problem and also in decentralized problem. Constraints (11) state that production units must at least produce enough material to satisfy the demand by the end of the time horizon.

$$\sum_{(jst,n)} outflow1(jst,n) + \sum_{(j,n)} outflow2(s,j,n) \ge d(s), \quad \forall jst\in Jst(s), j\in J, n\in N, s\in S \tag{11}$$

Duration Constraints:

<u>Time Sequence Constraints for Each Unit</u>: Constraints (12) to (14) express that if task (i) starts at event point ($n+1$), then it must start after the end of the same task happening at event point (n) in the same unit (j). If task (i) takes place at unit (j) at event point (n) then $wv(i,j,n)=1$, and $Ts(i,j,n+1)$ must be greater than or equal to $Ts(i,j,n)$. If $wv(i,j,n)=0$ then the constraint in equation (12) is relaxed and constraints in equations (13) and (14) enforce the sequencing of tasks.

$$Ts(i,j,n+1) \geq Tf(i,j,n) - UH(1-wv(i,j,n)), \quad \forall i \in I,\ j \in J(i), n \in N \tag{12}$$

$$Ts(i,j,n+1) \geq Ts(i,j,n), \quad \forall i \in I,\ j \in J(i),\ n \in N \tag{13}$$

$$Tf(i,j,n+1) \geq Tf(i,j,n), \quad \forall i \in I,\ j \in J(i),\ n \in N \tag{14}$$

Constraints (15) represent the rule that if task (i') should happen at event point (n) in unit (j) then task (i) must start at event point ($n+1$) after the end of task (i') at event point (n).

$$Ts(i,j,n+1) \geq Tf(i',j,n) - UH(1-wv(i',j,n)), \quad \forall j \in J(i), i \in I(j), i' \in I(j), i \neq i', n \in N \tag{15}$$

Constraints (16) to (19) represent that two consecutive productions, where unit (j) consumes the material produce by unit (j'), with no storage in between, happen at the same time. If $wv(i,j,n)$=1 and $wv(i',j',n)$=1 then $Ts(i,j,n)=Ts(i',j',n)$ and $Tf(i,j,n)=Tf(i',j',n)$. If either $wv(i,j,n)$=0 or $wv(i',j',n)$=0, then the constraints are relaxed.

$$\begin{gathered} Ts(i,j,n) \leq Ts(i',j',n) + UH(2 - wv(i,j,n) - wv(i',j',n)), \\ \forall j' \in J,\ j \in Jseq(j'),\ i \in I(j),\ i' \in I(j'),\ n \in N \end{gathered} \tag{16}$$

$$\begin{gathered} Ts(i,j,n) \geq Ts(i',j',n) - UH(2 - wv(i,j,n) - wv(i',j',n)), \\ \forall j' \in J,\ j \in Jseq(j'),\ i \in I(j),\ i' \in I(j'),\ n \in N \end{gathered} \tag{17}$$

$$\begin{gathered} Tf(i,j,n) \leq Tf(i',j',n) + UH(2 - wv(i,j,n) - wv(i',j',n)), \\ \forall j' \in J,\ j \in Jseq(j'),\ i \in I(j),\ i' \in I(j'),\ n \in N \end{gathered} \tag{18}$$

$$\begin{gathered} Tf(i,j,n) \geq Tf(i',j',n) - UH(2 - wv(i,j,n) - wv(i',j',n)), \\ \forall j' \in J,\ j \in Jseq(j'),\ i \in I(j),\ i' \in I(j'),\ n \in N \end{gathered} \tag{19}$$

<u>Time Sequence Constraints Connecting Unit and Storage Tank</u>: Constraints (20) to (23) state that production and storage occur at the same time. If unit (j) produces the material that is stored in tank (jst) then start and finishing time of production task at unit (j) and inflow to the storage tank must be same.

$$\begin{gathered} Ts(i,j,n) \leq Tss(j,jst,n) + UH(2 - wv(i,j,n) - in(j,jst,n)), \\ \forall jst \in Jst,\ j \in Jprodst(jst),\ i \in I(j),\ n \in N \end{gathered} \tag{20}$$

$$\begin{gathered} Ts(i,j,n) \geq Tss(j,jst,n) - UH(2 - wv(i,j,n) - in(j,j_{st},n)), \\ \forall jst \in Jst,\ j \in Jprodst(jst),\ i \in I(j),\ n \in N \end{gathered} \tag{21}$$

$$Tf(i,j,n) \le Tsf(j,jst,n) + UH(2 - wv(i,j,n) - in(j,jst,n)),$$
$$\forall jst \in Jst,\ j \in Jprodst(jst),\ i \in I(j),\ n \in N \tag{22}$$

$$Tf(i,j,n) \ge Tsf(j,jst,n) - UH(2 - wv(i,j,n) - in(j,jst,n)),$$
$$\forall jst \in Jst,\ j \in Jprodst(jst),\ i \in I(j),\ n \in N \tag{23}$$

Constraints are given by Equations (24) to (27) state that storage and production happen at the same time. If tank (*jst*) stores the material that is consumed in the unit (*j*), then outflow from storage tank and production take place at the same time.

$$Ts(i,j,n) \le Tss(jst,j,n) + UH(2 - wv(i,j,n) - out(j_{st},j,n)),$$
$$\forall jst \in Jst,\ j \in Jstprod(jst),\ i \in I(j),\ n \in N \tag{24}$$

$$Ts(i,j,n) \ge Tss(jst,j,n) - UH(2 - wv(i,j,n) - out(j_{st},j,n)),$$
$$\forall jst \in Jst,\ j \in Jstprod(jst),\ i \in I(j),\ n \in N \tag{25}$$

$$Tf(i,j,n) \le Tsf(jst,j,n) + UH(2 - wv(i,j,n) - out(jst,j,n)),$$
$$\forall jst \in Jst,\ j \in Jstprod(jst),\ i \in I(j),\ n \in N \tag{26}$$

$$Tf(i,j,n) \ge Tsf(jst,j,n) - UH(2 - wv(i,j,n) - out(j_{st},j,n)),$$
$$\forall jst \in Jst,\ j \in Jstprod(jst),\ i \in I(j),\ n \in N \tag{27}$$

<u>Sequence Constraints for Storage Tank</u>: Constraints are given in Eqns. (28) to (35) connect input and output flow happening at event (*n*) to the next event point (*n+1*) for storage tanks. Input starts at event point (*n+1*) after output finishes at event point (*n*) and output flow happens at event point (*n+1*) after the end of input flow happening at event point (*n*). Binary variable *in(j,jst,n)* equal to 1 when material is flowing into the tank (*jst*) from unit (*j*) at event point (*n*), otherwise its zero. Similarly *out(jst,j,n)* equal to 1 when material is flowing from the tank to the unit at event point (*n*), otherwise its zero.

Constraints (28), (31), (34), and (35) are active when the binary variables are equal to 1, whereas when the binary variables are equal to 0, the sequence constraints are enforced by Eqns. (29) , (30), (32), and (33).

$$Tss(j,jst,n+1) \ge Tsf(j,jst,n) - UH(1 - in(j,jst,n)), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ n \in N \tag{28}$$

$$Tss(j,jst,n+1) \ge Tss(j,jst,n), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ n \in N \tag{29}$$

$$Tsf(j, jst, n+1) \geq Tsf(j, jst, n), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ n \in N \tag{30}$$

$$Tss(jst, j, n+1) \geq Tsf(jst, j, n) - UH(1 - out(jst, j, n)), \quad \forall jst \in Jst,\ j \in Jstprod(jst),\ n \in N \tag{31}$$

$$Tss(jst, j, n+1) \geq Tss(jst, j, n), \quad \forall jst \in Jst,\ j \in Jstprod(jst),\ n \in N \tag{32}$$

$$Tss(jst, j, n+1) \geq Tsf(jst, j, n), \quad \forall jst \in Jst,\ j \in Jstprod(jst),\ n \in N \tag{33}$$

$$\begin{aligned} &Tss(j', jst, n+1) \geq Tsf(jst, j, n) - UH(1 - out(jst, j, n)), \\ &\qquad \forall jst \in Jst,\ j \in Jstprod(jst),\ j' \in Jprodst(jst),\ n \in N \end{aligned} \tag{34}$$

$$\begin{aligned} &Tss(jst, j, n+1) \geq Tsf(j', jst, n) - UH(1 - in(j', jst, n)), \\ &\qquad \forall jst \in Jst,\ j \in Jstprod(jst),\ j' \in Jprodst(jst),\ n \in N \end{aligned} \tag{35}$$

As we mentioned in section 3.2, there are two types of operating scenario for storage tanks and for each one of them, we have the following additional constraints.

Scenario 1: Simultaneous Loading and Unloading Not Allowed

Scenario 1 represents that material cannot flow in and out of the storage tank at the same time at any event point (*n*). This operation rule is used in many refineries for security reasons. Constraints (36) represent that output flow from storage tank (*jst*) starts after input ends at any event point (*n*).

$$\begin{aligned} &Tsf(j, jst, n) - UH(1 - in(j, jst, n)) \leq Tss(jst, j', n) + UH(1 - out(jst, j', n)), \\ &\qquad \forall jst \in Jst,\ j \in Jprodst(jst),\ j' \in Jstprod(jst),\ n \in N \end{aligned} \tag{36}$$

Scenario 2: Simultaneous Loading and Unloading Allowed

The operation rule for scenario 2 is that material can flow in and out of tank at the same time at any time interval (*n*). This assumption is very common in many refineries for intermediate storage tanks.

Constrains (37) to (40) are enforced when both variables *in(j,jst,n)* and *out(jst,j,n)* are equal to 1. When the constraints are enforced, the starting and finishing times of loading and unloading events are equal. When either *in(j,jst,n)* or *out(jst,j,n)* is equal to zero, the constraints are relaxed.

$$Tss(j, jst, n) + UH(1 - in(j, jst, n)) \geq Tss(jst, j', n) - UH(1 - out(jst, j', n)), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ j' \in Jstprod(jst),\ n \in N \tag{37}$$

$$Tss(j, jst, n) - UH(1 - in(j, jst, n)) \leq Tss(jst, j', n) + UH(1 - out(jst, j', n)), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ j' \in Jstprod(jst),\ n \in N \tag{38}$$

$$Tsf(j, jst, n) + UH(1 - in(j, jst, n)) \geq Tsf(jst, j', n) - UH(1 - out(jst, j', n)), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ j' \in Jstprod(jst),\ n \in N \tag{39}$$

$$Tsf(j, jst, n) - UH(1 - in(j, jst, n)) \leq Tsf(jst, j', n) + UH(1 - out(jst, j', n)), \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ j' \in Jstprod(jst),\ n \in N \tag{40}$$

Makespan Constraints:

Constraints (41) to (46) state that starting and finishing time of any task is always less than equal to the makespan (*H*).

$$Ts(i, j, n) \leq H, \quad \forall i \in I,\ j \in J(i),\ n \in N \tag{41}$$

$$Tf(i, j, n) \leq H, \quad \forall i \in I,\ j \in J(i),\ n \in N \tag{42}$$

$$Tss(j, jst, n) \leq H, \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ n \in N \tag{43}$$

$$Tsf(j, jst, n) \leq H, \quad \forall jst \in Jst,\ j \in Jprodst(jst),\ n \in N \tag{44}$$

$$Tss(jst, j, n) \leq H, \quad \forall jst \in Jst,\ j \in Jstprod(jst),\ n \in N \tag{45}$$

$$Tsf(jst, j, n) \leq H, \quad \forall jst \in Jst,\ j \in Jstprod(jst),\ n \in N \tag{46}$$

Objective Function:

Finally, equation (47) defines the objective function of the problem which is the minimization of makespan. The most common motivation for optimizing the process using minimization of makespan as objective function is to improve customer services by accurately predicting order delivery dates.

$$z = \min(H) \tag{47}$$

3.4. Solution Approach

The CPU time for the solution of the overall model presented in the previous section is usually high due to model size. In order to reduce the CPU resolution time we developed a structural decomposition strategy which decomposes the problem into a number of smaller and thus easier to solve subsystems. The developed structural decomposition approach and the additional constraints presented in this section guarantee that the solution obtained by the decentralized optimization will be feasible for the centralized system and is exactly the same.

Structural Decomposition Approach

Generally scheduling problems are large scale problems and are difficult to be solved to optimality. As the scale of the production increases, the mathematical complexity of the developed model increases, and the CPU time that is required for the solution of the corresponding problem increases too. Decomposition is a natural way to dealing with large scale problems. There are two types of decomposition: the structural decomposition of the system under study and the mathematical decomposition such as Benders decomposition (Benders, 1962), Lagrangian relaxation (Minoux, 1986) etc. Note that mathematical decomposition can be applied after the application of the structural decomposition if it is applicable.

The decomposition strategy proposed here decomposes the refinery scheduling problem presented in section 3.2 spatially. To obtain the optimal solution in decentralized optimization approach, each sub-system is solved to optimality and these optimal results are used to obtain the optimal solution for the entire problem. In our proposed decomposition rule, we split the system in such a way so that a minimum amount of information is shared between the sub-problems. This means splitting the problem at intermediate storage tanks such that the inflow and outflow streams of the tank belong to different sub-systems. The decomposition starts with the final products or product storage tanks, and continues to include the reactors/units that are connected to them and stops when the storage tanks are reached. The products, intermediate products, units and storage tanks are part of the sub-system 1. Then following the input stream of each storage tank, the same procedure is used to determine the next sub-system. If input and output stream of the tank are included at the same local problem then the storage tank also belongs to that local problem.

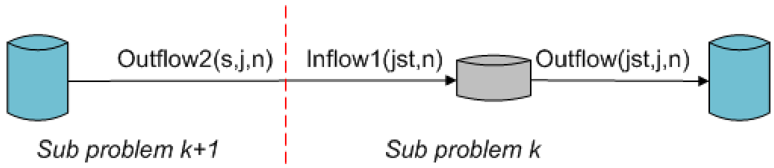

Figure 3.2 Intermediate storage tank connecting two sub-systems

When the problem is decomposed at intermediate storage tanks, storage tanks become a connecting point between two sub-systems. The amount and type of material flowing out of the connecting intermediate storage tank at any time interval (*n*) becomes demand for the preceding sub-system (*k+1*) at corresponding time interval (see Figure 3.2).

After decomposing the centralized system, the individual sub-systems are treated as independent scheduling problems and solved to optimality using the mathematical formulation described in section 3.3. It should be also noticed that the operating rules for the decentralized system are the same as those required for the centralized problem. In general the local optimization of sub-system k gives minimum information to the sub-system k+1 which optimizes its schedule with the restrictions regarding the demand of the intermediates obtained by sub-system k. In Figure 3.3 we present the decomposition of the system under study after the application of the developed decomposition rule. The system is split in two sub-systems where sub-system 1 produces all of the final products and one by-product. The sub-system 1 includes 5 production unit, 7 final product storage tanks, and 3 raw material tanks, Raw material tanks in sub-system 1 are defined as intermediate tanks in centralized system. The sub-system 2 includes 4 production units, 1 intermediate tank, 2 raw material tanks and it produces 4 final products. Except Coke, all other final products in sub-system 2 are defined as intermediate products in centralized system.

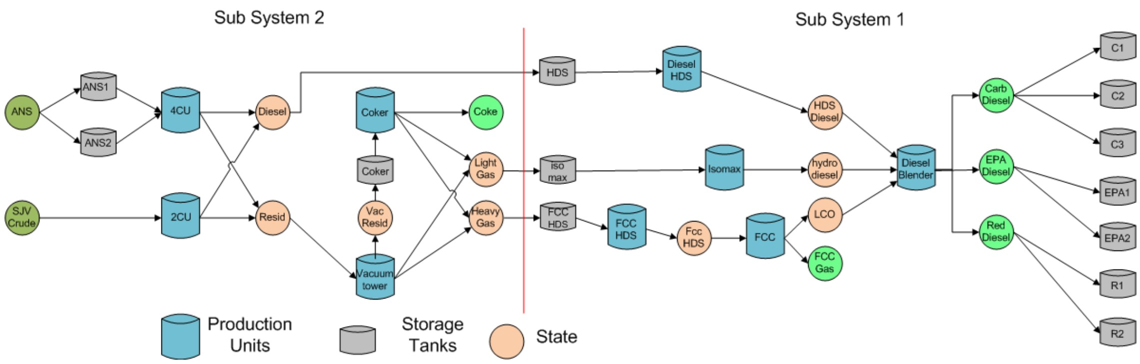

Figure 3.3 Decomposition of the refinery problem under consideration

In the following section we present all the additional constraints used in the decentralized approach in order to guarantee that the obtained solution by the sub-system 1 could be realized by the sub-system 2.

<u>Feasibility Constraints for Decentralized Model</u>

The sub-systems obtained using the decomposition rule presented in the previous section, have all the constraints presented in the basic model but in addition to that the *k+1* sub-system has to satisfy the demand of final products produced by this sub-system and also the demand of intermediate products needed by sub-system *k.* The demand constraints for intermediate final products for sub-system *k+1* are given by equation (48).

$$\sum_j outflow2(s,j,n) \geq r(s,n), \quad \forall s \in S, j \in Junitp(s,k+1), n \in N \tag{48}$$

Scenario 1: Simultaneous Loading and Unloading Not Allowed

When production units in sub-system *k+1* supply material to storage tanks located in sub-system *k*, in order to obtain globally feasible solution, the following capacity constraints are added to sub-system *k+1.* Constraint in equation (49) is for time interval *n*=0; sum of the material supplied to storage tank (*jst*) in sub-system *k* and initial amount present in the storage tank must be within tank capacity limit. Whereas equations (50) and (51) represents capacity constraints for event point *n*=1 and *n*=2 respectively.

$$\sum_{j} outflow2(s,j,n) + stin(jst) \le V^{\max}(jst), \quad \forall s \in S,\ jst \in Jst(s,k),\ j \in Junitp(s,k+1), n = 0 \tag{49}$$

$$\sum_{j}\sum_{n=0}^{1} outflow2(s,j,n) + stin(jst) - r_s(0) \le V^{\max}(jst), \\ \forall s \in S,\ jst \in Jst(s,k),\ j \in Junitp(s,k+1), n \in N \tag{50}$$

$$\sum_{j}\sum_{n=0}^{2} outflow2(s,j,n) + stin(jst) - \sum_{n=0}^{1} \mathrm{r(s,n)} \le V^{\max}(jst), \\ \forall s \in S,\ jst \in Jst(s,k),\ j \in Junitp(s,k+1), n \in N \tag{51}$$

Constraints (52) and (53) represent lot sizing constraints for sub-system *k+1*. The demand of intermediate final product *s* at event point *n* is adjusted by the amount present in the storage tank after the demand is satisfied at previous event point (*n-1*). This adjusted demand is then used in demand constraints for intermediate final products.

$$r(s,1) - \left(\sum_{j} outflow2(s,j,0) + stin(jst) - r(s,0) \right) = r'(s,1), \\ \forall s \in S,\ j \in Junitp(s,k+1),\ jst \in Jst(s,k) \tag{52}$$

$$r(s,2) - \left(\sum_{j}\sum_{n=0}^{1} outflow2(s,j,n) + stin(jst) - \sum_{n=0}^{1} r(s,n) \right) = r'(s,2), \\ \forall s \in S,\ j \in Junitp(s,k+1),\ jst \in Jst(s,k), n \in N \tag{53}$$

The optimal time horizon of global problem is obtained by combining the optimal schedules of sub-systems at each point (*n*) such that the material balance constraints are satisfied for connecting intermediate storage tanks. Since sub-system *k+1* satisfies the demand of sub-system *k*, sub-system *k+1* will happen before the sub-system *k*.

Scenario 2: Simultaneous Loading and Unloading Allowed

To implement the constraint of simultaneous loading and unloading to intermediate connecting storage tanks between any two sub-systems *k* and *k+1*, the following constraints are added to sub-system *k*.

$$in\mathrm{flow1}(jst,n) = \beta(s) \times \left(Tsf(jst,j,n) - Tss(jst,j,n) \right), \quad \forall s \in S,\ jst \in Jst(s,k), j \in Junitc(s), n \in N \tag{54}$$

$$\beta(s) = P_{\max}^{k+1}(s) \quad if \ \ P_{\max}^{k+1}(s) < C_{\max}^{k}(s), \quad \forall s \in S \tag{55}$$

$$\beta(s) = C^{k}_{max}(s) \quad \text{if } C^{k}_{max}(s) < P^{k+1}_{max}(s), \quad \forall s \in S \tag{56}$$

where $P^{k+1}_{max}(s)$ is the maximum rate of production of intermediate final product (*s*) in sub-system *k+1* and $C^{k}_{max}(s)$ is the maximum rate of consumption of (*s*) in sub-system *k*. $P^{k+1}_{max}(s)$ and $C^{k}_{max}(s)$ can be calculated before the start of the optimization process based on the configuration of sub-systems. β(s) takes the value of either the maximum rate of production or maximum rate of consumption as given by constraints (55) and (56). Constraint (54) determines the value of material inflow to the storage tank at event point (*n*). When β(s) takes the value of maximum rate of consumption, constraint (57) is added to the sub-system *k+1*.

$$\sum_{j} outflow2(s, j, n) = \beta(s) \times (Tf(i, j, n) - Ts(i, j, n)), \quad \forall s \in S, j \in Junitp(s, k+1), i \in I(j), n \in N \tag{57}$$

Constraints (57) state that the total amount of material (*s*) which is an intermediate final product, produced in sub-system *k+1* is equal to the maximum rate of consumption in sub-system *k* times duration of production in sub-system *k+1*.

$$inflow1(jst, n) = iter \times \sum_{j} outflow2(s, j, n), \quad \forall s \in S, jst \in Jst(s, k), j \in Junitp(s, k+1), n \in N \tag{58}$$

where *iter* is a binary variable. After solving sub-systems *k* and *k+1* to optimality respectively, if the optimal makespan results in a time horizon of sub-system *k+1 to be* greater than sub-system *k*, then *iter* = 1. When *iter* = 1, we resolve sub-system *k* to optimality with constraint (58) active in the model. Eq. (58) specifies the amount of material flowing into the connecting intermediate storage tank from sub-system *k+1* at each event point (*n*).

Since, loading and unloading can happen at the same time, the time horizon of each sub-system will be the same and the global optimal solution can be obtained by combining the optimal schedule of each sub-system.

3.5. Numerical Results

The refinery production scheduling case study presented here is based on realistic data provided by Honeywell Hi-Spec Solutions. The data for the problem studied here are presented in

Appendix Chapter 3. Different demand cases for final products, Carb diesel, EPA diesel, and Red Dye diesel; and residual products, Coke and FCC gas, are studied. The actual values of the products' demands are given in Table A3-1. For all computations in this chapter GAMS/CPLEX 7.0 is used to solve the resulting MILP formulations. The optimal solution is obtained with 1e-6 integrality gap using a Pentium(R) 4 processor at 3.40Hz and with 1.99GB memory. The two scenarios presented in section 3.3, with and without the assumption of simultaneously loading and unloading of tank are examined. Centralized and decentralized optimization is applied using the decomposition approach presented in section 3.4. In the following tables four different examples are presented in order to illustrate the advantage of decentralized optimization using the decomposition approaches. Four examples represent lower to high demands for the system that need to be satisfied within available time horizon of 240 hours.

Following the mathematical model presented is section 3.3, the 1st sub-system as shown in Figure 3.3 is solved to minimize the makespan and meet the demand for the final products. Based on the optimal solution of sub-system 1, sub-system 2 is solved to optimality such that it satisfies the demand required by sub-system 1. In the end the solutions of the two sub-systems are combined to obtain the solution of the entire problem.

As shown in Table 3.1 and Table 3.2, after the application of the decomposition strategy the size of subsystems is significantly reduced (ex. scenario 1 from 1081 to 749) giving rise to a small increase in the number of constraints. This happens because the decision variables associated with connecting the two sub-systems in centralized problem become demand data in decentralized approach resulting to additional constraint to the first sub-system.

Table 3.1 Characteristics of mathematical formulation using Scenario 1

	Continuous Variables	Binary Variables	Total number of variables	Constraints	Total number of constraints

Centralized	985	96	1081	1425	1425
Sub System 1	484	54	749	921	1686
Sub System 2	187	24		765	

Table 3.2 Characteristics of mathematical formulation using Scenario 2

	Continuous Variables	**Binary Variables**	**Total number of variables**	**Constraints**	**Total number of constraints**
Centralized	985	96	1081	1488	1425
Sub System 1	484	54	743	930	1485
Sub System 2	181	24		555	

To obtain the global optimal solution for scenario 1, the optimal schedules of each sub-system are combined at each event point *n* such that the material balance and storage capacity constraints for intermediate connecting tanks are satisfied, whereas the global optimal schedule in scenario 2 is obtained by superimposing the optimal solution of each sub-system. As shown in Table 3.3 and Table 3.4 for both scenarios, the centralized and decentralized optimizations give exactly the same optimal makespan for all examples. As explained in the following section, sub-system 1 has exactly the same solution in centralized and decentralized optimization which gives rise to the same optimal solution (centralized and decentralized) for sub-system 2.

The objective function in centralized and decentralized strategy is minimization of makespan. In order to spend minimum time producing material (*s*), it is required to operate all the units in the system in such a way that they will produce all materials needed at a maximum production rate that is feasible for that particular system. The maximum possible production rate ($P_{max}(s)$) for each product (s) can be determined based on the data given in Table A3-2, Table

A3-3. Storage tank capacity data are given in Table A3-4. In sub-system 2, only one unit (*j*) produces the specific material (*s*), which means that $P_{\max}(s)$ is defined only by the highest feasible production rate of the unit (*j*). This highest feasible production rate of unit (*j*) is calculated by taking under consideration the units that come before and after unit (*j*) (without any storage in between). Thus, the highest feasible production rate for unit (*j*) depends only on the parameters and the configuration of the system therefore, it is constant in both centralized and decentralized approach for all event points (*n*). We can then conclude that the total time spent in order to produce (*s*) is the same in centralized and decentralized system and this gives rise to identical makespan for sub-system 1. Subsequently, given that the makespan for sub-system 1 is the same in centralized and decentralized system, the sub-system 2 takes the same demand in centralized and decentralized optimization because the final product demand requirements are the same in both optimization approaches. Since, sub-system 2 has the same configuration, same data, and same constraints in centralized and decentralized system, then sub-system 2 has the same optimal solution (may be with different task assignments) concerning the makespan in both optimization approaches.

As shown in Table 3.3 and Table 3.4, the CPU time required to find the optimal makespan is reduced significantly in decentralized approach compared to centralized approach. This is mainly due to the reduction in size and complexity of the system going from centralized system to sub-systems.

Table 3.3 Scenario 1 Centralized vs. Decentralized

Ex.	**Centralized System**		**Decentralized System**				
	CPU Time (s)	Objective Value (hr)		CPU-Time(s)	Total CPU-time(s)	Local Objective value(hr)	Global Objective(hr)

1	318.672	235.890	Sub 1 Sub 2	0.969 2.313	3.282	97.000 138.89	235.890
2	256.187	163.352	Sub 1 Sub 2	1.141 2.921	4.062	72.00 91.352	163.352
3	2192.515	79.328	Sub 1 Sub 2	1.063 2.984	4.047	36.960 42.368	79.328
4	589.984	26.149	Sub 1 Sub 2	1.093 2.218	3.311	10.272 15.878	26.149

Table 3.4 Scenario 2 Centralized vs. Decentralized

Ex.	**Centralized System**		**Decentralized System**				
	CPU Time (s)	**Objective Value (hr)**		**CPU-Time(s)**	**Total CPU-time(s)**	**Local Objective value(hr)**	**Global Objective(hr)**
1	4.125	235.000	Sub 1 Sub 2	1.031 0.641	1.672	235.000 235.000	235.000
2	5.844	215.000	Sub 1 Sub 2	1.047 0.766	1.813	215.000 215.000	215.000
3	5.562	191.000	Sub 1 Sub 2	1.031 1.312	2.343	191.000	191.000
4	7.562	97.000	Sub 1 Sub 2	1.359 1.188	2.547	97.000 97.000	97.000

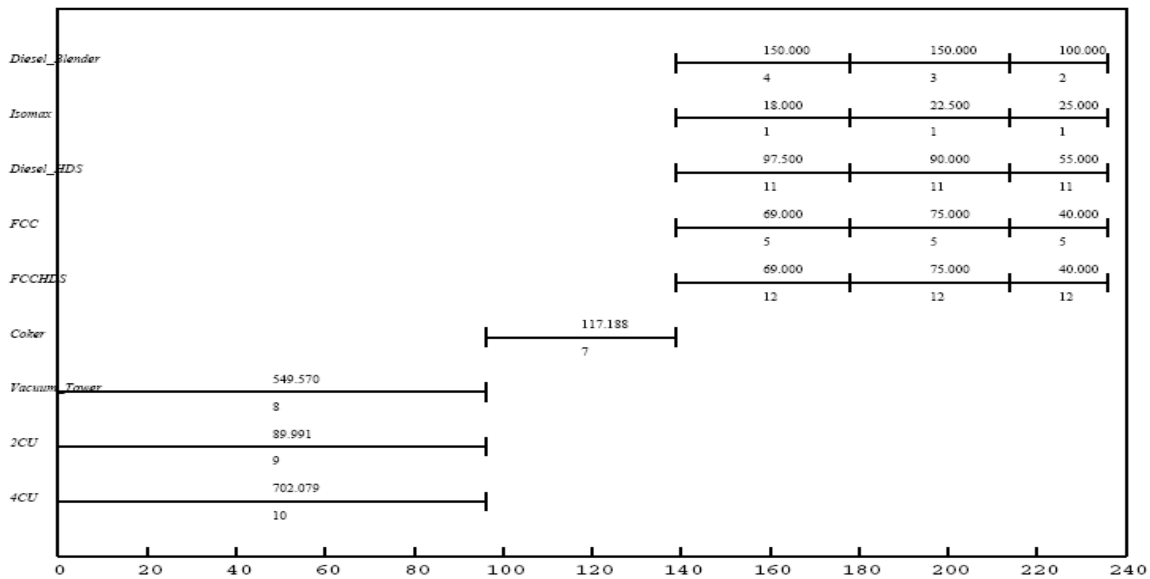

Figure 3.4 Gantt chart of the operation schedule for example 1, centralized system, scenario 1.

For scenario 1, the CPU time needed to solve the centralized system is in the order of 100 seconds whereas the decentralized system is solved within 5 CPU seconds. In scenario 2, decentralized solution approach also show improvement compared to centralized system. The CPU time needed to solve the problem is cut by half in decentralized system compared to centralized system. For scenario 1, the production schedule obtained by decentralized approach is different than that obtained by centralized approach as shown in Figure 3.4 and Figure 3.5 for example 1. This difference in production schedule is obtained because in decentralized system, an optimal solution is obtained by integrating the schedules of each sub-system at each event point. Gantt charts for storage tanks are given in Figure A3-1 to Figure A3-10 for centralized and decentralized systems and for scenario 2.

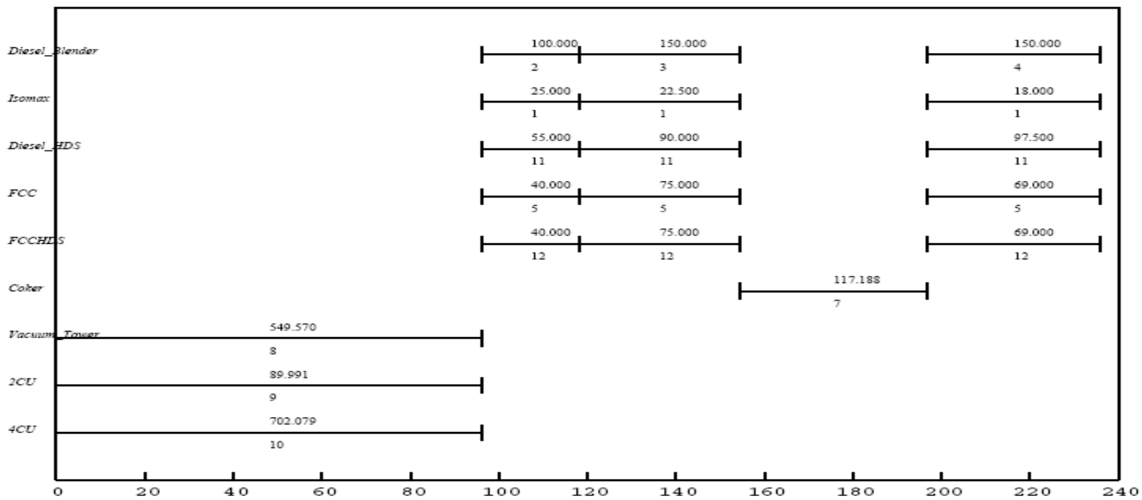

Figure 3.5 Gantt chart of the operation schedule for Case 4, decentralized system, scenario 1.

3.6. Summary

In this chapter, a structure decomposition strategy and formulation is presented for short-term scheduling of refinery operations. It is shown that the decentralized system model results in fewer constraints and fewer continuous and binary variables compared to centralized system. The chapter presents a problem where both optimization strategies result in the same optimal makespan with the advantage of having significant reduction in the computational time for decentralized system compared to that of centralized system. For decisions making process in refinery where demands always need to be satisfy and determination of the makespan is the main concern, proposed approach provides a decomposition scheme that provides global optimal solution in decentralized system. Furthermore, the proposed decomposition approach is suitable for scheduling of production unit operations because they need to satisfy blend components demand required by blend scheduling operations in time.

Nomenclature

Indices

j	Production units
jst	Storage tanks

i	Tasks
n	Event points
s	States
k	k^{th} sub-system in decentralized system

Sets

J	Production units
Jst	Storage tanks
S	States
N	Event point within the time horizon
$J(i)$	Units which are suitable for performing task *i*
$I(j)$	Tasks which can be performed in unit *j*
$Iseq(i')$	i' produces state *s* that will be consumed by task i
$Jstprod(jst)$	Units that consume material *s* stored in tank *jst*
$Jprodst(jst)$	Units that produce material *s* stored in tank *jst*
$Junitp(s)$	Units that can produce material *s*
$Junitp(s,k)$	Units in sub-system *k* that can produce material *s*
$Junitc(s)$	Units that consume material *s*
$Junitc(s,k)$	Units in sub-system *k* that can consume material *s*
$Jseq(j')$	Units that follow unit j' (no storage in between)
$Jst(s)$	Tanks that can store material *s*
$Jst(s,k)$	Tanks in sub-system *k* that can store material *s*
$JSTprodst(j)$	Tanks that follow unit *j*
$JSTstprod(j)$	Tanks that are followed by unit *j*

Parameters

$R^{\min}(i,j)$	Minimum rate of material processed by task i required to start production unit j
$R^{\max}(i,j)$	Maximum rate of material processed by task i in unit j
$P_{\max}(s)$	The possible maximum rate of production of material (s)
$P_{\max}^{k}(s)$	The maximum rate of production of intermediate final product s in sub-system k
$C_{\max}^{k}(s)$	The maximum rate of consumption of intermediate final product s in sub-system k
$V^{\max}(jst)$	Maximum available storage capacity of storage tank jst
$\rho^{p}(s,i)$	Proportion of state s produced by task i, $\rho^{p}(s,i) \geq 0$
$\rho^{c}(s,i)$	Proportion of state s consumed by task i, $\rho^{c}(s,i) \geq 0$
$d(s)$	Demand of the final product s at the end of the time horizon
$r(s,n)$	Demand of intermediate state s at event point n
$r'(s,n)$	Adjusted demand of intermediate state s at event point n
$stin(jst)$	Amount of state s that is present at the beginning of the time horizon
UH	Available time horizon

Variables

$wv(i,j,n)$	Binary variable that assign the starting of task i in unit j at event point n
$iter$	Binary variable that assign the number of iterations between sub-systems
$b(i,j,n)$	Amount of material undertaking task i in unit j at event point n
$st(jst,n)$	Amount of state s present in storage tank jst at event point n

$inflow1(jst,n)$	Flow of raw material to storage tank *jst* event point *n*
$outflow1(jst,n)$	Flow of final product from storage tank *jst* at event point *n*
$inflow2(s,j,n)$	Flow of raw material *s* to production unit *j* at point *n*
$outflow2(s,j,n)$	flow of product material *s* from unit *j* at point *n*
$inflow(j,jst,n)$	Flow of material from unit *j* to storage tank *jst* event point *n*
$outflow(jst,j,n)$	Flow of material from storage tank *jst* to unit *j* at point *n*
$in(j,jst,n)$	Binary variable that assign the starting of material flow into storage tank *jst* from unit *j* at point *n*
$out(jst,j,n)$	Binary variable that assign the starting of material flow out of storage tank *jst* to unit *j* at point *n*
$unitflow(s,j,j',n)$	Flow of state *s* from unit *j* to consecutive unit *j'* for consumption at point *n*
$st(jst,n)$	Amount of material in tank *jst* at event point *n*
$Ts(i,j,n)$	Time that task *i* starts in unit *j* at event point *n*
$Tf(i,j,n)$	Time that task *i* finishes in unit *j* at event point *n*
$Tss(j,jst,n)$	Time that material starts to flow from unit *j* to storage tank *jst*
$Tsf(j,jst,n)$	Time that material finishes to flow from unit *j* to tank *jst* at event point *n*
$Tss(jst,j,n)$	Time that material starts to flow from tank *jst* to unit *j* at event point *n*
$Tsf(jst,j,n)$	Time that material finishes to flow from tank *jst* to unit *j* at event point *n*
H	Time horizon

Chapter 4

4. Scheduling of a large-scale oil-refinery operations

Refineries are increasingly concerned with improving the scheduling of their operations in order to achieve better economic performances by minimizing quality, quantity, and logistics give away. In this chapter, we present a comprehensive integrated optimization model based on continuous-time formulation for the scheduling problem of production units and end-product blending problem. The model incorporates quantity, quality, and logistics decisions related to real-life refinery operations. These involve minimum run-length requirements, fill-draw-delay, one-flow out of blender, sequence dependent switchovers, maximum heel quantity, and downgrading of better quality product to lower quality. The logistics giveaways in our work are associated with obtaining a feasible solution while minimizing violations of sequence dependent switchovers and maximum heel quantity restrictions. A set of valid inequalities are proposed that improves the computational performance of the model significantly. The formulation is used to address realistic case studies where feasible solutions are obtained in reasonable computational time.

4.1. Introduction

The oil-refinery production operation is one of the most complex chemical industries, which involves many different and complicated processes with various connections. Over the last 20 years, it has grown increasingly more complex due to tighter competition, stricter environment regulations, and lower margin profits. Main objective of the oil-refineries is to transform crude-oil into gasoline, diesel, jet fuel, and other middle distillate hydrocarbon products that can be used as either feedstock or energy source in chemical process industry. The short-term scheduling is a critical aspect in this large and complex production process. The refinery operations scheduling problem involves decisions that are related to quantity, quality and logistics. Quantity decisions include lot-sizes for raw material, intermediate and product tanks inventories, amount of material moving between production units and storage tanks, etc, while quality decisions deal with obtaining finished products that meet specific quality requirements. Logistics constraints include policies and procedures for production operations which deal with allocating resources to operations, sequencing or ordering of different modes of operations, and determining the durations of operations. To remain competitive in dynamic global marketplace, the oil-refineries are increasingly concerned with improving the scheduling of their operations in order to achieve better economic performance by minimizing product quality violations, incidence of high quality product giveaway, and logistics giveaway. Oil-refinery operations can be classified into three sub-operations based on the structure of the refinery configuration as shown in Figure 4.1. (Jia et al., 2003) These sub-operations are (1) crude-oil unloading and mixing, (2) production unit operations, and (3) finished products blending and distribution. Sub-operation 1 involves the crude-oil unloading, blending, and inventory control, Sub-operation 2 includes production unit scheduling, and Sub-operation 3 consists of finished product blending and lifting. Typically, sub-operations scheduling optimization problems are addressed separately since centralized optimization approach gives rise to an incomprehensive large scale mixed integer non-linear programming (MINLP) models. Review of refinery scheduling problem has been presented in work of N. K. Shah et al. (2011).

The crude-oil unloading and mixing scheduling problem has been studied by Pedro M. Castro and Grossmann (2014), Lee et al. (1996), Jia et al. (2003) , Kolodziej et al. (2013), Reddy et al. (2004a), Reddy et al. (2004b), G. K. D. Saharidis, Minoux, et al. (2009), and G. K. D. Saharidis et al. (2010). The complex crude-oil blend scheduling optimization problem is decomposed into logistics and quality sub-problems by Kelly and Mann (2003a, 2003b). They utilized successive linear programming (SLP) to solve the quality sub-problem. Once the crude prepared mixture is prepared by crude oil loading and unloading operations, it is charged to CDUs for distillation. The distillation cuts from CDU are then sent to other production units for fractionation and reaction to produce blend components for finished products. The most common refinery includes catalytic, hydro, and thermal cracking units to convert heavy hydrocarbons into light hydrocarbons. They also include other process units like continuous catalytic reforming, hydro treating, and hydro desulfurization units. The production unit operations scheduling problem is characterized by continuous processes, intermediate component storage, and recycle stream.

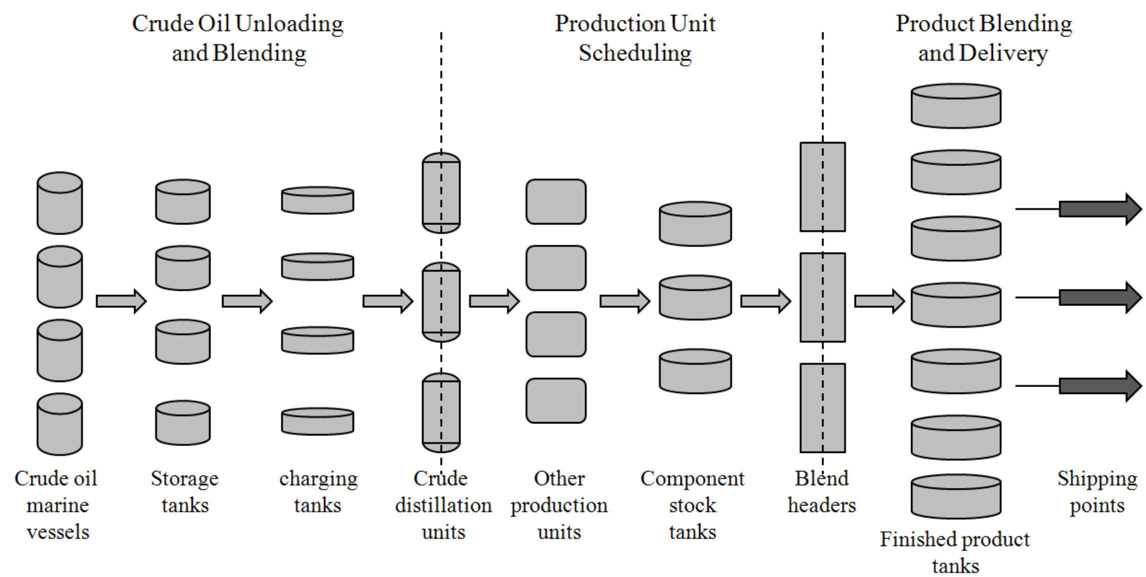

Figure 4.1 Graphic overview of a standard refinery system

The production units scheduling problem based on continuous time representation is proposed by Joly et al. (2002) and Jia and Ierapetritou (2004) where they applied spatial decomposition to solve the scheduling problem. Gary et al. (2007) is a comprehensive reference for refinery production unit operations. The finished products blend scheduling has received significant attention in the literature. The purpose of blend scheduling optimization problem is to find the best way of mixing different semi-finished products that have been rectified during various refinery processes with some additives so as to produce final products that meet quality specifications and demand while minimizing cost. The gasoline blending operation is highly non-linear and gives rise to a mixed integer non-linear programing scheduling model. Blend recipe optimization problems considering nonlinear properties relations have been studied extensively.

(Castillo & Mahalec, 2014a, 2014b; Cuiwen et al., 2013; Glismann & Gruhn, 2001; J. Li & Karimi, 2011; J. Li et al., 2010; Mendez et al., 2006; Zhao et al., 2008) Glismann and Gruhn (2001) proposed a decomposition technique based on first solving the nonlinear (NLP) quality optimization model and then solving a MILP model to optimize temporal and resource decisions. J. Li and Karimi (2011) developed a slot-based MILP formulation for an integrated treatment of recipe, specifications, blending, and storage considering real-life features such as multipurpose product tanks, parallel nonidentical blenders, minimum run lengths, changeovers, piecewise constant profiles for blend component qualities and feed rates, etc. To reduce the computational complexity that arises from non-linear model, many have used constant component properties. (Jia & Ierapetritou, 2003; Joly et al., 2002) Kelly (2006) emphasized the importance of logistics details in refinery blending and delivery problem and proposed a decomposition of the blend scheduling problem into two sub-problems, logistics and quality. The logistics sub-problem considers only the quantity and logistics related variables and the problem constraints whereas the quality sub-problem considers product specifications, quantity constraints and bounds. Their work is based on discrete time representation and their formulation includes many logistics details such as minimum run-length, sequence dependent changeovers, and fill-draw-delay. They observed that incorporating logistics details into scheduling problem can yield substantial improvements in efficiency and productivity. A MILP optimization model based on continuous time representation, using unit specific event points, and fixed blend recipe was developed by Jia and Ierapetritou (2003). They modeled multipurpose product tanks, but they do not include certain features such as sequence dependent switchovers constraints, fill-draw-delay at product tank, and one flow out of blender. An iterative procedure is proposed by Mendez et al. (2006) to deal with variable recipe and nonlinear properties for different grades of products by replacing MINLP with sequential MILP formulations. They enforced prepared blend recipe whenever it is possible. In their model formulation they did not consider multipurpose product tanks, fill-draw-delay, and minimum run-length requirement. There is extensive literature on the refinery blending problem using nonlinear optimization tools. (Adhya et al., 1999; Audet et al., 2004; Gounaris et al., 2009)

With more product grades, stricter specifications, new government regulations, and fewer feasible blends, refineries face increasing challenges to maintain, let alone increase, profitability. What is needed is a comprehensive model for the entire refinery scheduling problem and for various refinery configurations that addresses quality, quantity, and logistics issues in a unified, yet flexible approach. The comprehensive model is complex, hard to build and solve and there is sparse work in literature in this area. Moro et al. (1998) proposed a planning model for refinery diesel production where the emphasis is on blending relations. Pinto et al. (2000) proposed a planning and scheduling model for refinery production and distribution operations. Their formulation is based on discretization of time and the model includes features such as sequence dependent transition cost of products within oil pipeline. C. Luo and Rong ; Chunpeng Luo and Rong (2007) developed a two-tiered decision-making hierarchical scheduling model for overall refinery. The upper level optimization model is based on discrete time formulation and it is used to determine sequencing and timing of operations modes and to decide the quantities of materials produced/consumed at each operations mode. The upper decisions level uses aggregated tanks storage capacity whereas the lower level uses heuristics to obtain a detailed schedule. They consider multipurpose product tanks via an iterative procedure that allows to readjust aggregated tank capacity at the lower level by changing multipurpose tank service mode and then to recalculate corresponding optimal solution at the upper level. The logistics details are ensured through heuristics at the lower level. There are also several commercial tools available for refinery scheduling such as Aspen Petroleum Scheduler, Aspen Refinery Multi-Blend Optimizer, Honeywell’s Production Scheduler, and Honeywell’s Blend. Honeywell’s Production Scheduler has a logistics solver to optimize logistics and quantity problems and a quality solver to solve quantity and quality problems. Most of the models utilized in these commercial tools are based on discrete time formulation.

In this work, we present a comprehensive integrated optimization model for the production units and end-product blend scheduling problem that incorporates quantity, quality and logistics decisions related to real-life refinery operations. The model is driven by the shipment plan and accounts for the tradeoffs between costs of keeping inventory and changing run-modes. The goals of refinery operations scheduling are to maximize the profit and performance while minimizing the penalties subject to quantity, quality, and logistics giveaways and nonattainment. (Kelly, 2003a) The outline of the chapter is as follow. Section 4.2 presents the problem definition, section 4.3 presents the mathematical formulation which is applied to a realistic case study example to illustrate the applicability of proposed model to large scale model in section 4.4, and the chapter concludes with section 4.5.

4.2. Problem Definition

The production scheduling determines the detailed schedule of each production unit and each demand order unloading for a short time period (typically 10 days - 1 month) by taking into account the operational constraints of the plant. The schedule defines which products should be produced and which materials should be consumed in each time interval over a given small time horizon; hence, it defines which run-mode to use and when to perform changeovers in order to meet the market needs satisfying the demand and product specifications. Large-scale scheduling problems arise frequently in oil refineries where the main objective is to assign sequence of tasks to processing units within certain time frame such that the demand of each product is satisfied before its due date while minimizing the cost or maximizing the total profit.

The refinery production system considered here is composed of raw material storage tanks, production units, blending units, intermediate tanks, and final product tanks. Each production unit is defined as a continuous processing element that transforms the input streams into several products according to the variable production recipe. For reasons of operating flexibility and cost effectiveness, refinery unit operations can generate a range of intermediate streams which are blended into finished products. For simplicity, we separate the final products into two groups: a) products that are stored in tanks and b) products that are not stored in tanks but are supplied to the market directly from production units. In this work, we limit the use of term "demand order" only for the first group of products and each demand order corresponds to only one kind of product since each multiproduct order can be decomposed into several single product orders. The characteristics of the problem considered in this chapter are given in detail in the next subsection.

Problem Characteristics

The key information available for the refinery include the following:

1) Maximum and minimum proportion of material produced or consumed at each production unit
2) Maximum and minimum production flow-rates for each production unit
3) Maximum and minimum inventory capacities for each storage tank, identity of material type that each tank can service, and initial holdup in each tanks
4) Upper limit on the flows out of finished product tanks
5) Demand orders for group A products and their delivery time windows
6) Total demand for group B products
7) Maximum allowable heel quantity for multipurpose product tanks
8) Minimum run-length for units and maintenance time for multipurpose tanks between run-modes
9) Quality specifications limit on blend product properties
10) Available scheduling time horizon of 10 days

The goal of optimization is to determine:

1) The sequencing of tasks for production units

2) Each product pool that satisfies demand orders

3) Durations of tasks at production units and duration of unloading tasks from product pools

4) The inventory levels in component and product pools

5) Production rates for units and unloading rate for product pools

6) Composition of material produced and consumed

The problem is also restricted by a series of logistics details as follows:

1) At any given time, only one task can take place at production unit

2) Minimum run-length constraint for production units

3) Non-contiguous product order fulfillment for product in group A

4) Blend unit can send product to multiple pool sequentially, not simultaneously

5) Product tanks cannot distribute and receive material at the same time

6) Fill-draw-delay restriction for product pools enforces certain amount of downtime on tanks after product loading event has taken place

7) Multipurpose tanks can store different types of materials over time, but only one type of material at any given time

8) Sequence dependent switch over for multipurpose tanks where higher quality product is stored before lower grade products

9) Maximum heel requirement restriction does not allow product heel to exceed specified maximum heel quantity when the multipurpose tank is switched to a different mode

10) Downgrading of the higher grade product to lower grade product if necessary

Although a realistic case study is modeled the following assumptions had to be made:

1) Unlimited supply of raw materials

2) Fixed recipe for crude distillation units (CDUs)

3) Constant blend components properties

4) Perfect mixing in the blender

5) Product tank cannot satisfy multiple demand orders simultaneously

6) Each demand order involves only one product

7) For production units, the amount of time required for run-modes change is neglected

8) Changeover times in multipurpose tanks from higher to lower grade product are negligible

Before presenting the mathematical model, a case study with realistic data provided by Honeywell Process Solutions (HPS) is presented in the next section. The refinery produces diesel fuels, jet fuel, and components for gasoline production. This case study will be used to illustrate the applicability of the proposed model in section 4.3.

4.2.1. Case study description

The production process at Honeywell refinery consists of 2 blender units, 13 other processing units, and 2 non-identical parallel crude distillation units (CDUs) that process two different type of crude oils. The schematic of production system is shown in Figure 4.2. There are two charging tanks for one CDU and a charging pipeline for another CDU. The CDUs concurrently transform crude oil into several distillation cuts. These distillation cuts from the CDUs are then sent to other production units for fractionation and reaction to produce blend components for finished products. The oil-refinery under case study has following production units: Vacuum Tower, Coker conversion unit, Continuous catalytic reforming (CCR) process unit, Isomax, Fluid catalytic cracking (FCC) unit, Penex, De-C5, and Alkylation process unit. Current regulatory requirements to produce ultra-low-sulfur fuels require the use of hydrotreating technology. Thus, the refinery also includes three hydrodesulfurization (HDS) units: Naphtha HDS, Diesel HDS, fluid catalytic cracking HDS (FCC HDS).

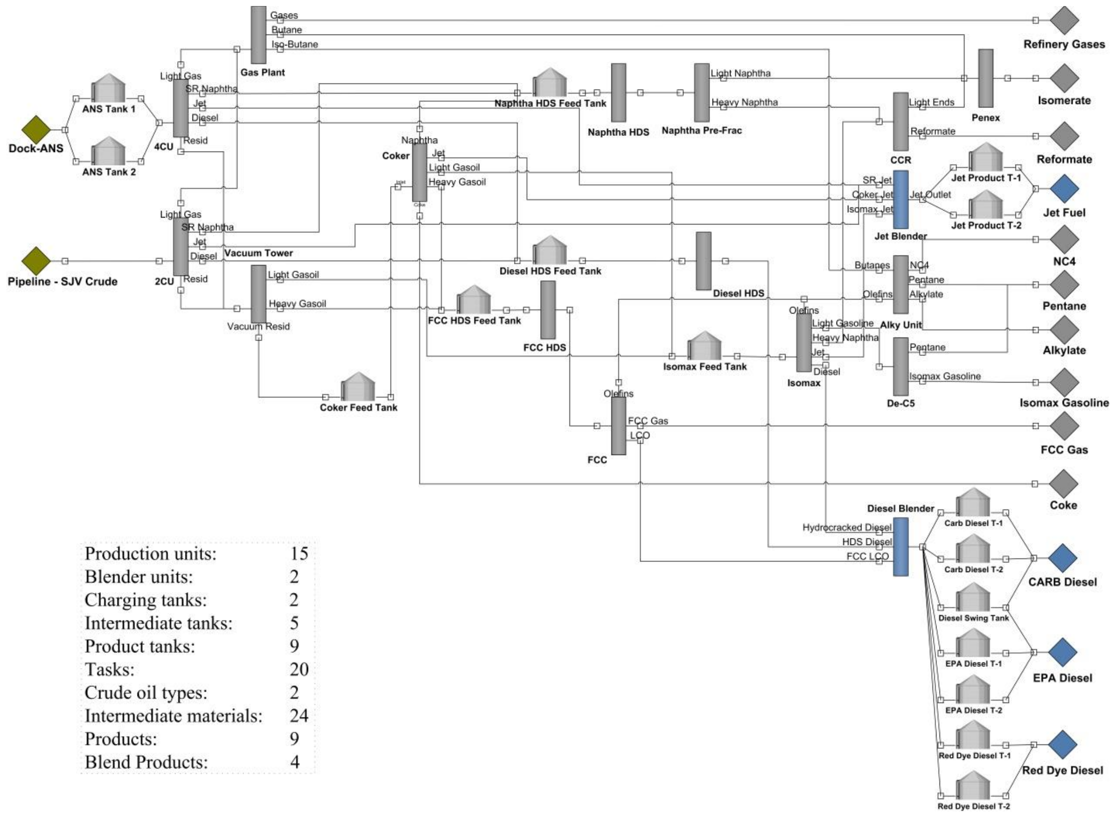

Figure 4.2 Schematic of the refinery production plant for reference example

Honeywell refinery utilizes Jet blender and Diesel blender units to blend components produced by other production units to produce final products. Jet blender unit blends straight run jet, coker jet, and isomax jet streams to produce jet fuel which can be stored in 2 product tanks. Diesel blender unit produces three different grades of fuel: CARB diesel, EPA diesel, and Red-dye diesel. Light cycle oil, HDS diesel, and hydrocracked diesel are blended to produce these three grades of diesel products using three different run-modes. There are two dedicated tanks for each grade of diesel products and one multipurpose tank that can service CARB and EPA diesel.

There is 6 hours of cleaning or maintenance downtime when the multipurpose tank service switches from lower grade of diesel product to higher grade of product. This cleaning downtime is essential to remove any sulfur contamination present in the tank before low sulfur product is sent for storage. The product tank has 4 hours of down time called fill-draw-delay for certificate of analysis preparation and to let the product settle down and mix before is shipped to the market.

4.3. Mathematical Formulation

In this section we present the mathematical formulation for the refinery production scheduling problem based on continuous time representation and the idea of unit specific event points.(M. G. Ierapetritou & C. A. Floudas, 1998; M.G. Ierapetritou & C.A. Floudas, 1998; Ierapetritou et al., 1999) A state-task network (STN) representation introduced by Kondili et al. (1993) is used to describe the refinery operations. The model involves material balance constraints, capacity constraints, demand constraints, quality constraints, logistics constraints, and setup constraints. Material balance constraints connect the amount of material at one event point to next event point, storage and production capacity limit is enforced by capacity constraints, and demand constraints ensure that all the products demand is satisfied while quality constraints ensure product quality specifications. Logistics constraints include all the logistics details presented in the previous section. If a feasible solution that satisfies all the quantity, quality, and logistics constraints cannot be obtained, then it is essential to produce a schedule that can still be implemented in real-life refinery sacrificing model feasibility. In this case we introduced artificial variables to treat any infeasibility present and these variables are subsequently penalized in the objective function to obtain an optimal solution that satisfies as many as possible from the quantity, logistics, and quality constraints by minimizing giveaways. A detailed description of the each variable and parameter used in the model can be found in the nomenclature section.

Variable recipe constraints:

Upper and lower bounds are forced on the individual components volumetric flow rates processed at each production units. Here $B_{i,j,n}$ is a total amount of material processed at unit j performing task i at event point n. Constraint (1a) enforces the bound for the amount produced, whereas constraint (1b) ensures that the amount consumed is restricted by the imposed recipe.

$$\rho_{s,i}^{p,\min} B_{i,j,n} \le bp_{s,i,j,n} \le \rho_{s,i}^{p,\max} B_{i,j,n}, \quad \forall s \in S, i \in I_s^P, j \in J_i, n \in N \tag{1a}$$

$$\rho_{s,i}^{c,\min} B_{i,j,n} \le bc_{s,i,j,n} \le \rho_{s,i}^{c,\max} B_{i,j,n}, \quad \forall s \in S, i \in I_s^C, j \in J_i, n \in N \tag{1b}$$

Furthermore, the amount of material processed is equal to the total amount of material consumed or produced. Therefore, constraints (1a-1b) can be replaced by (2a-2c) and we can eliminate variable $B_{i,j,n}$. Constraint (2c) satisfies material balance at each production unit j, which states that the total amount of material consumed is equal to the total amount of material produced.

$$\rho_{s,i}^{p,\min} \sum_{s' \in S_i^p} bp_{s',i,j,n} \le bp_{s,i,j,n} \le \rho_{s,i}^{p,\max} \sum_{s' \in S_i^p} bp_{s',i,j,n}, \quad \forall s \in S, i \in I_s^P, j \in J_i, n \in N \tag{2a}$$

$$\rho_{s,i}^{c,\min} \sum_{s' \in S_i^c} bc_{s',i,j,n} \le bc_{s,i,j,n} \le \rho_{s,i}^{c,\max} \sum_{s' \in S_i^c} bc_{s',i,j,n}, \quad \forall s \in S, i \in I_s^C, j \in J_i, n \in N \tag{2b}$$

$$\sum_{s \in S_i^p} bp_{s,i,j,n} = \sum_{s \in S_i^c} bc_{s,i,j,n}, \quad \forall j \in J, i \in I_j, n \in N \tag{2c}$$

In our work we assume that the crude distillation units have the same lower and upper bounds which means that the distillation cuts are assumed to be known.

Material balance constraints for production units:

Constraints (3a) connect the material produced at production units to subsequent storage tanks, production units, and end product delivery to market. Constraints (3b) represents that the consumption at a production unit is equal to the amount of material coming from preceding storage tanks, previous units, and raw material supply.

$$\sum_{i \in I_j} bp_{s,i,j,n} = \sum_{k \in K_j^{pk} \cap K_s} Kif_{s,j,k,n} + \sum_{j' \in J_j^{seq} \cap J_s^c} JJf_{s,j,j',n} + Uof_{s,j,n}, \quad \forall s \in S, j \in J_s^p, n \in N \tag{3a}$$

$$\sum_{i \in I_j} bc_{s,i,j,n} = \sum_{k \in K_j^{kp} \cap K_s} Kof_{s,k,j,n} + \sum_{j' \in J_j^{seq} \cap J_s^p} JJf_{s,j',j,n} + Uif_{s,j,n}, \quad \forall s \in S, j \in J_s^c, n \in N \tag{3b}$$

Material balance constraints for storage tanks:

The material balance constraints for storage tanks are given by equations (4a-4b). The equations state that the inventory of a tank at one event point is equal to that of previous event point adjusted by the input and output streams amount and by taking into account the downgraded products amount.

$$\begin{aligned} st_{s,k,n} = sto_{s,k} + \sum_{j \in J_k^{pk}} Kif_{s,j,k,n} + Rif_{s,k,n} - \sum_{j \in J_k^{kp}} Kof_{s,j,k,n} - \sum_{o \in O_s} Lf_{k,o,n} \\ + \sum_{s' \in S_k} std_{s',s,k,n} - \sum_{s' \in S_k} std_{s,s',k,n}, \quad \forall s \in S, k \in K_s, n = 1 \end{aligned} \tag{4a}$$

$$\begin{aligned} st_{s,k,n} = st_{s,k,n-1} + \sum_{j \in J_k^{pk}} Kif_{s,j,k,n} + Rif_{s,k,n} - \sum_{j \in J_k^{kp}} Kof_{s,j,k,n} - \sum_{o \in O_s} Lf_{k,o,n} \\ + \sum_{s' \in S_k} std_{s',s,k,n} - \sum_{s' \in S_k} std_{s,s',k,n}, \quad \forall s \in S, k \in K_s, 1 < n \le N \end{aligned} \tag{4b}$$

The variable $std_{s',s,k,n}$ is defined as the tank heel of material s' present at the end of event point $n-1$ that is downgraded to material s during event point n.

$$std_{s',s,k,n} = \begin{cases} \ge 0 & \text{if product } s' \text{ is downgraded to } s \text{ in tank } k \text{ at event point } n \\ 0 & \text{otherwise} \end{cases}$$

When there is a changeover from higher grade product to lower grade, the tank heel present in the tank would be transformed into lower grade product without violating any product property specifications of lower grade product.

Capacity constraints for production units:

Constraint (5) enforces that the material processed by unit j performing task i is bounded by the maximum and minimum rate of production. Constraint (6) gives an upper bound on total amount of material processed at unit j over the entire time horizon.

$$R_{i,j}^{\min}\left(Tf_{i,j,n} - Ts_{i,j,n}\right) \le \sum_{s' \in S_i^p} bp_{s',i,j,n} \le R_{i,j}^{\max}\left(Tf_{i,j,n} - Ts_{i,j,n}\right), \quad i \in I, j \in J_i, n \in N \tag{5}$$

$$\sum_{s' \in S_i^p} bp_{s',i,j,n} \le UH \times R_{i,j}^{\max} \times wv_{i,j,n}, \quad \forall i \in I, j \in J_i, n \in N \tag{6}$$

Capacity constraints for storage tanks:

Constraints (7a-7c) are capacity constraints for storage tanks and they define the binary variables associated with flow in and out of the tanks.

$$Kif_{s,j,k,n} \le V_k^{\max} \times in_{s,j,k,n}, \quad \forall j \in J, k \in K_j^{pk}, n \in N \tag{7a}$$

$$Kof_{s,j,k,n} \le V_k^{\max} \times out_{s,j,k,n}, \quad \forall j \in J, k \in K_j^{kp}, n \in N \tag{7b}$$

$$Lf_{s,k,o,n} \le V_k^{\max} \times l_{k,o,n}, \quad \forall s \in S, k \in K_s, o \in O_s, n \in N \tag{7c}$$

Constraints (8a-8c) represent that the material present in the tank should not exceed maximum storage capacity. The multipurpose tanks can store different grades of products and if the higher grade product is present in the tank when it is being serviced for lower grade product, then the high quality product will be downgraded into lower quality. The downgraded products are taken into consideration for storage capacity limit in constraints (8a-8b).

$$sto_{s,k} + \sum_{j \in J_k^{pk}} Kif_{s,j,k,n} + Rif_{s,k,n} + \sum_{s' \in S_k} std_{s',s,k,n} - \sum_{s' \in S_k} std_{s,s',k,n} \le V_k^{\max} y_{s,k,n}, \quad \forall s \in S, k \in K_s, n = 1 \tag{8a}$$

$$st_{s,k,n-1} + \sum_{j \in J_k^{pk}} Kif_{s,j,k,n} + Rif_{s,k,n} + \sum_{s' \in S_k} std_{s',s,k,n} - \sum_{s' \in S_k} std_{s,s',k,n} \le V_k^{\max} y_{s,k,n}, \quad \forall s \in S, k \in K_s, 1 < n \le N \tag{8b}$$

$$\sum_{j \in J_k^{kp}} Kof_{s,k,j,n} + \sum_{o \in O_s} Lf_{k,o,n} \le V_k^{\max} y_{s,k,n}, \quad \forall s \in S, k \in K_s, n \in N \tag{8c}$$

The maximum and minimum unloading (lift) rate for product storage tanks must be bounded as specified by constraint (9).

$$RU_k^{\min}\left(Tof_{k,o,n} - Tos_{k,o,n}\right) \le Lf_{k,o,n} \le RU_k^{\max}\left(Tof_{k,o,n} - Tos_{k,o,n}\right), \quad \forall k \in K_p, o \in O, n \in N \tag{9}$$

Quality constraints:

The final products produced by the blenders should satisfy the quality specifications. These product qualities are assumed to be computed by volumetric average to maintain model linearity. Since the blend components properties are assumed to be constant, linearity of the model is preserved. Constraint (10) guarantees that final product leaving the outlet port of the blender satisfies set product quality range. Here, $P_{s,p}^{\min}$ and $P_{s,p}^{\max}$ are the upper and lower limit of property *p* for final blend product *s*.

$$P_{s,p}^{\min}\sum_{i\in I_s^p} bp_{s,i,j,n} - pg_{s,p,n}^{l} \le \sum_{i\in I_s^p, s'\in S_i^c} P_{s',p} bc_{s',i,j,n} \le P_{s,p}^{\max}\sum_{i\in I_s^p} bp_{s,i,j,n} + pg_{s,p,n}^{u}, \quad \forall s\in S_b, j\in J_s^p, n\in N \tag{10}$$

When the final products produced by the blenders cannot meet the quality specifications at event point *n*, we introduced positive slack variables $pg_{s,p,n}^{l}$ and $pg_{s,p,n}^{u}$ which are penalized in objective function to minimize giveaways.

Demand constraints:

Demand of each finished product must be satisfied during the entire scheduling horizon. Constraint (11) guarantees that sufficient amount of product will be available to meet the demand.

$$D_{o,s}^{-} + r_s - dg_o^{l} - rg_s + \le \sum_{k,n} Lf_{k,o,n} + \sum_{j,n} Uof_{s,j,n} \le D_{o,s}^{+} + dg_o^{u}, \quad \forall s\in S_b, o\in O_s \cup \forall s\in S_f \tag{11}$$

Due to production capacity limitation, sometimes the demand order of finished product cannot be satisfied during the entire scheduling horizon. A feasible solution can be obtained by introducing the positive artificial variables dg_o^l, dg_o^u and rg_s in the demand constraint which are penalized in the objective function to minimize quantity giveaway.

Logistics Constraints:

Allocation constraints: Constraint (12) expresses that if a task i starts at event point n, then it must be performed in one of the suitable units j. It also satisfies the operating detail that a unit can physically perform only one task at any given time.

$$\sum_{i\in I_j} wv_{i,j,n} \le 1, \quad \forall j\in J, n\in N \tag{12}$$

The requirement that multipurpose storage tanks can store only one type of products at any time is enforced by equation (13).

$$\sum_{s\in K_s} y_{s,k,n} \le 1, \quad \forall k\in K^m, n\in N \tag{13}$$

Minimum run-lengths: The minimum run-length for each task is enforced by constraint (14). Here the minimum run-length (RL_i) is 15 hours.

$$Tf_{i,j,n} - Ts_{i,j,n} \geq RL_i wv_{i,j,n} - UH\left(1 - wv_{i,j,n}\right), \quad \forall i \in I, j \in I_j, n \in N \tag{14}$$

Similar to minimum run-length constraint, maximum run-length restriction can be imposed if necessary.

Loading and unloading constraints: Product tanks cannot load and unload material at the same time and this restriction is enforced by equations (15). Furthermore, we restrict that product tanks can only satisfy one demand order at any given time.

$$\sum_{s \in S_k} in_{s,j,k,n} + \sum_{o \in O} l_{k,o,n} \leq 1, \quad \forall k \in K, j \in J_k^{pk}, n \in N \tag{15}$$

A one-flow-out restriction for blender unit: A one-flow-out restriction given by equation (16) is required for all operation tasks on a blender to ensure that the product output from the blend unit can only go to one product tank. This restriction is imposed because the refinery has a blend property online controller that is set up to fill a specific product tank by taking into account the tank heel properties. If a blend unit does not comply with this restriction and send the output from the blender to multiple tanks simultaneously and not sequentially, then this miss-operation can result in a significant off-specification of product stocks.

$$\sum_{k \in K_s \cap K_j^{pk}} in_{s,j,k,n} \leq 1, \quad \forall s \in S_b, j \in J_s^p, n \in N \tag{16}$$

Set-up constraints: To model tank set-up, we use the binary variables $\beta_{k,n}$ which are defined by constraints (17a-17b).

$$\beta_{k,n} = \begin{cases} 1 & \text{if storage tank } k \text{ becomes active at event point } n \text{ for first time} \\ 0 & \text{otherwise} \end{cases}$$

$$\beta_{k,n} \leq \sum_{s \in S_k} y_{s,k,n} + \sum_{s \in S_k, n' < n} y_{s,k,n'}, \quad \forall k \in K^h, n \in N, \sum_s yo_{s,k} = 0 \tag{17a}$$

$$\beta_{k,n} \geq \sum_{s \in S_k} y_{s,k,n} - \sum_{s' \in S_k, n' < n} y_{s',k,n'}, \quad \forall k \in K^h, n \in N, \sum_s yo_{s,k} = 0 \tag{17b}$$

Similarly for production unit, the setup variables are $\alpha_{j,n}$ and constraints (18a-18b) represent the utilization of unit at event n for the very first time during the production time horizon.

$$\alpha_{j,n} = \begin{cases} 1 & \text{if unit } j \text{ becomes active at event point } n \text{ for first time} \\ 0 & \text{otherwise} \end{cases}$$

$$\alpha_{j,n} \leq \sum_{i \in J_i} wv_{i,j,n} + \sum_{i \in J_i, n' < n} wv_{i,j,n'}, \quad \forall j \in J^h, n \in N \tag{18a}$$

$$\alpha_{j,n} \geq \sum_{i \in J_i} wv_{i,j,n} - \sum_{i' \in J_{i'}, n' < n} wv_{i',j,n'}, \quad \forall j \in J^h, n \in N \tag{18b}$$

Since, the refinery operates in a continuous mode; we only define the setup variables for identical parallel production units and tanks. Setup variables are penalized in the objective function to minimize the total number of units and storage tanks that are utilized during refinery operation.

Changeovers constraints: Different mode of production and storage tasks are specified by different types of product being produced or stored. Changeovers between modes of operations cause disturbances and additional costs. Thus, few changeovers (long sequences of the same mode of operations) are desired. Continuous variables $\chi_{i',i,j,n}$ and $\eta_{s',s,k,n}$ denote changeover of task at production unit j and changeover of service mode at product pool k, respectively.

$$\chi_{i',i,j,n} = \begin{cases} 1 & \text{if mode changes from } i' \text{ at event point } n \text{ to } i \text{ at later event point} \\ 0 & \text{otherwise} \end{cases}$$

$$\eta_{s',s,k,n} = \begin{cases} 1 & \text{if mode changes from } s' \text{ at event point } n \text{ to } s \text{ at later event point} \\ 0 & \text{otherwise} \end{cases}$$

Changeover constraints proposed by Shaik et al. (2009) are used in this work. Constraints (19-20c) are thus used to force the changeover variables to 1 if there is a change in operations mode from event point n to any later event point.

$$\chi_{i',i,j,n} \leq wv_{i',j,n}, \quad \forall j \in J^m, i \in I_j, i' \in I_j, i' \neq i, n < N \tag{19a}$$

$$\chi_{i',i,j,n} \le 1 + wv_{i,j,n'} - \sum_{i''\in I_j} wv_{i'',j,n'} + \sum_{i''\in I_j, n''\in n<n''<n'} wv_{i'',j,n''}, \quad \forall j \in J^m, i \in I_j, i' \in I_j, i' \ne i, n < N, n < n' \le N \tag{19b}$$

$$\chi_{i',i,j,n} \ge wv_{i',j,n} + wv_{i,j,n'} - 1 - \sum_{i''\in I_j, n<n''<n'} wv_{i'',j,n''}, \quad \forall j \in J^m, i \in I_j, i' \in I_j, i' \ne i,, n < N, n < n' \le N \tag{19c}$$

$$\eta_{s',s,k,n} \le y_{s',k,n}, \quad \forall k \in K^m, s \in S_k, s' \in S_k, s' \ne s, n < N \tag{20a}$$

$$\eta_{s',s,k,n} \le 1 + y_{s,k,n'} - \sum_{s''\in S_k} y_{s'',k,n'} + \sum_{s''\in S_k, n''\in n<n''<n'} y_{s'',k,n''}, \quad \forall k \in K^m, s \in S_k, s' \in S_k, s' \ne s, n < N, n < n' \le N \tag{20b}$$

$$\eta_{s',s,k,n} \ge y_{s',k,n} + y_{s,k,n'} - 1 - \sum_{s''\in S_k, n<n''<n'} y_{s'',k,n''}, \quad \forall k \in K^m, s \in S_k, s' \in S_k, s' \ne s,, n < N, n < n' \le N \tag{20c}$$

The changeover variable $\eta o_{s',s,k}$ in equation (20d) is active if there is material *s'* present in the tank at the beginning of the time horizon and then service is changed over to new material *s* at event point *n=1*.

$$yo_{s,k} + y_{s',k,n} \le \eta o_{s,s',k} + 1, \quad \forall k \in K^m, s \in S_k, s' \in S_k, s' \ne s, yo_{s,k} = 1 \tag{20d}$$

Changeovers variables $\chi_{i',i,j,n}, \eta o_{s',s,k}$ and $\eta_{s',s,k,n}$ are penalized in the objective function to minimize the changeovers.

Heel requirement: When changeover occurs from higher to lower quality product, the holdup in the tank must be less than equal to the maximum heel quantity specified. The maximum heel requirement can be enforced on $std_{s,s',k,n}$ as soft constraint using equations (21a-21b) and positive slack variable $mh_{s,k,n}$. The artificial variable $mh_{s,k,n}$ is penalized in the objective function to minimize the heel.

$$std_{s,s',k,n} - mh_{s,k,n} \le V_k^{heel} + V_k^{\max}\left(1 - \eta o_{s,s',k}\right), \quad \forall k \in K, s \in K_s, s' \in K_s, \varphi_{s'} < \varphi_s, n = 1 \tag{21a}$$

$$std_{s,s',k,n+1} - mh_{s,k,n+1} \le V_k^{heel} + V_k^{\max}\left(1 - \eta_{s,s',k,n}\right), \quad \forall k \in K, s \in K_s, s' \in K_s, \varphi_{s'} < \varphi_s, n < N \tag{21b}$$

Product downgrading: The downgrading of product happens, 1) there is a changeover of service at used multipurpose tanks or 2) it is required to meet lower quality product lifting demand order due to production capacity limitation. The downgrading of product is captured by equations (22a-22e). The variable associated with downgrading $std_{s,s',k,n}$ is zero when the switchover occurs from lower to higher quality product.

$$std_{s,s',k,n} \le 0, \quad \forall k \in K^m, s \in K_s, s' \in K_s, \varphi_{s'} > \varphi_s, n \in N \tag{22a}$$

$$std_{s,s',k,n} \le V_k^{\max} \eta o_{s,s',k}, \quad \forall k \in K^m, s \in K_s, s' \in K_s, \varphi_{s'} < \varphi_s, n = 1 \tag{22b}$$

$$std_{s,s',k,n+1} \le V_k^{\max} \eta_{s,s',k,n}, \quad \forall k \in K^m, s \in K_s, s' \in K_s, \varphi_{s'} < \varphi_s, n < N \tag{22c}$$

$$\begin{gathered} std_{s,s',k,n} - V_k^{\max}\left(1-\eta o_{s,s',k}\right) \le sto_{s,k} \le std_{s,s',k,n} + V_k^{\max}\left(1-\eta o_{s,s',k}\right), \\ \forall k \in K^m, s \in K_s, s' \in K_s, \varphi_{s'} < \varphi_s, n = 1 \end{gathered} \tag{22d}$$

$$\begin{gathered} std_{s,s',k,n+1} - V_k^{\max}\left(1-\eta_{s,s',k,n}\right) \le st_{s,k,n} \le std_{s,s',k,n+1} + V_k^{\max}\left(1-\eta_{s,s',k,n}\right), \\ \forall k \in K^m, s \in K_s, s' \in K_s, \varphi_{s'} < \varphi_s, n < N \end{gathered} \tag{22e}$$

Timing Constraints:

Sequence Constraints for production units: Finishing time of any task must be greater than the starting time of that task, as represented by constraint (23a). Constraint (23b) expresses that if task i starts at event point $n+1$, then it must start after the end of the same task happening at event point n while equation (23c) enforces the time sequence constraint for different tasks happening in the same unit.

$$Tf_{i,j,n} \ge Ts_{i,j,n}, \quad \forall i \in I, j \in J_i, n \in N \tag{23a}$$

$$Ts_{i,j,n+1} \ge Tf_{i,j,n}, \quad \forall i \in I, j \in J_i, n \in N, n < N \tag{23b}$$

$$Ts_{i,j,n+1} \ge Tf_{i',j,n} - UH\left(1 - wv_{i',j,n}\right), \quad \forall j \in J, i \in I_j, i' \in I_j, i' \ne i, n \in N, n < N \tag{23c}$$

Constraints (24a-24d) represent that two consecutive productions with no storage in between, happen at the same time because production units operate as continuous processes. Here, unit j consumes the material produced by unit j'.

$$Ts_{i,j,n} \le Ts_{i',j',n} + UH\left(2 - wv_{i,j,n} - wv_{i',j',n}\right), \quad \forall j' \in J, j \in J_{j'}^{seq}, i \in I_j, i' \in I_{j'}, n \in N \tag{24a}$$

$$Ts_{i,j,n} \ge Ts_{i',j',n} - UH\left(2 - wv_{i,j,n} - wv_{i',j',n}\right), \quad \forall j' \in J, j \in J_{j'}^{seq}, i \in I_j, i' \in I_{j'}, n \in N \tag{24b}$$

$$Tf_{i,j,n} \le Tf_{i',j',n} + UH\left(2 - wv_{i,j,n} - wv_{i',j',n}\right), \quad \forall j' \in J, j \in J_{j'}^{seq}, i \in I_j, i' \in I_{j'}, n \in N \tag{24c}$$

$$Tf_{i,j,n} \ge Tf_{i',j',n} - UH\left(2 - wv_{i,j,n} - wv_{i',j',n}\right), \quad \forall j' \in J, j \in J_{j'}^{seq}, i \in I_j, i' \in I_{j'}, n \in N \tag{24d}$$

Sequence Constraints for storage tanks: Finishing time of the inlet, outlet transfer service has to be greater than or equal to the start time of that service. Constraints (25-27) enforce the sequence time requirement for movement transfer task from one event point to next event point for same unit-tank connection.

$$Tsf_{j,k,n} \ge Tss_{j,k,n}, \quad \forall k \in K, j \in J_k^{pk}, n \in N \tag{25a}$$

$$Tss_{j,k,n+1} \ge Tsf_{j,k,n}, \quad \forall k \in K, j \in J_k^{pk}, n \in N, n < N \tag{25b}$$

$$Tsf_{k,j,n} \ge Tss_{k,j,n}, \quad \forall k \in K, j \in J_k^{kp}, n \in N \tag{26a}$$

$$Tss_{k,j,n+1} \ge Tsf_{k,j,n}, \quad \forall k \in K, j \in J_k^{kp}, n \in N, n < N \tag{26b}$$

$$Tof_{k,o,n} \ge Tos_{k,o,n}, \quad \forall k \in K_p, o \in O, n \in N \tag{27a}$$

$$Tos_{k,o,n+1} \ge Tof_{k,o,n}, \quad \forall k \in K_p, o \in O, n \in N, n < N \tag{27b}$$

Start time sequence constraints for tanks receiving/sending material from/to multiple production units is given by constraints (28a) and (28b). Whereas constraints enforcing time sequence requirement for different type of demand orders satisfied by the tank is given equation (28c).

$$Tss_{j',k,n+1} \geq Tsf_{j,k,n} - UH\left(1 - in_{s,j,k,n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{pk}, j' \neq j, n \in N, n < N \tag{28a}$$

$$Tss_{k,j',n+1} \geq Tsf_{k,j,n} - UH\left(1 - out_{k,j,n}\right), \quad \forall k \in K, j \in J_k^{kp}, j' \in J_k^{kp}, j' \neq j, n \in N, n < N \tag{28b}$$

$$Tos_{k,o',n+1} \geq Tof_{k,o,n} - UH\left(1 - l_{k,o,n}\right), \quad \forall k \in K_p, o \in O, o' \in O, o' \neq O, n \in N, n < N \tag{28c}$$

Sequence constraint for material transfer in and out of intermediate tanks happening at the same event point is enforced by equations (29a-29d).

$$Tss_{j,k,n} + UH\left(1 - in_{s,j,k,n}\right) \geq Tss_{k,j',n} - UH\left(1 - out_{s,k,j',n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{kp}, n \in N \tag{29a}$$

$$Tss_{j,k,n} - UH\left(1 - in_{s,j,k,n}\right) \leq Tss_{k,j',n} + UH\left(1 - out_{s,k,j',n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{kp}, n \in N \tag{29b}$$

$$Tsf_{j,k,n} + UH\left(1 - in_{s,j,k,n}\right) \geq Tsf_{k,j',n} - UH\left(1 - out_{s,k,j',n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{kp}, n \in N \tag{29c}$$

$$Tsf_{j,k,n} - UH\left(1 - in_{s,j,k,n}\right) \leq Tsf_{k,j',n} + UH\left(1 - out_{s,k,j',n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{kp}, n \in N \tag{29d}$$

Constraints (30a-30b) connect material transfer from one event point to next event point. Product tanks cannot simultaneously load and unload material and this restriction is enforced by equations (31a-31b). Variable ta_k is a fill-draw-delay parameter for tank k. The tank unloading happens anytime after the material flow into the tank is over and fill-draw-delay downtime has elapsed. Product flow into the tank starts after the end of the product unloading.

$$Tss_{j,k,n+1} \geq Tss_{k,j',n} - UH\left(1 - out_{s,k,j',n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{kp}, n \in N, n < N \tag{30a}$$

$$Tss_{k,j',n+1} \geq Tss_{j,k,n} + UH\left(1 - in_{s,j,k,n}\right), \quad \forall k \in K, j \in J_k^{pk}, j' \in J_k^{kp}, n \in N \tag{30b}$$

$$Tos_{k,o,n+1} \geq Tsf_{j,k,n} - UH\left(1 - in_{s,j,k,n}\right) + ta_k in_{s,j,k,n}, \quad \forall s \in S, k \in K_s, j \in J_k^{pk}, o \in O_s, n < N \tag{31a}$$

$$Tss_{j,k,n+1} \geq Tof_{k,o,n} - UH\left(1 - l_{k,o,n}\right), \quad \forall k \in K, j \in J_k^{pk}, o \in O, n < N \tag{31b}$$

The tank needs to go through cleaning maintenance to store higher grade product after servicing lower grade product. This maintenance downtime requirement is captured by constraints (31c).

Constraints are relaxed if the there is no switchover in the service from lower grade to higher grade product type.

$$Tss_{j,k,n+1} \geq Tof_{k,o,n} + tclean_k \left(2\eta_{s',s,k,n} - 1\right), \quad \forall k \in K, s \in S_k, s' \in S_s, \varphi_{s'} < \varphi_s, j \in J_k^{pk}, o \in O_{s'}, n < N \tag{31c}$$

Sequence Constraints for production units and storage tanks: Upstream production and material flow into storage tank happens at the same time, which is imposed by constraints (32a-32d).

$$Ts_{i,j,n} \leq Tss_{j,k,n} + UH\left(2 - wv_{i,j,n} - in_{s,j,k,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{pk}, i \in I_j, n \in N \tag{32a}$$

$$Ts_{i,j,n} \geq Tss_{j,k,n} - UH\left(2 - wv_{i,j,n} - in_{s,j,k,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{pk}, i \in I_j, n \in N \tag{32b}$$

$$Tf_{i,j,n} \leq Tsf_{j,k,n} + UH\left(2 - wv_{i,j,n} - in_{s,j,k,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{pk}, i \in I_j, n \in N \tag{32c}$$

$$Tf_{i,j,n} \geq Tsf_{j,k,n} - UH\left(2 - wv_{i,j,n} - in_{s,j,k,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{pk}, i \in I_j, n \in N \tag{32d}$$

For intermediate storage tanks, downstream production and material flow out of feed tank occur simultaneously. This constraint is imposed by equations (33a-33d).

$$Ts_{i,j,n} \leq Tss_{k,j,n} + UH\left(2 - wv_{i,j,n} - out_{s,k,j,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{kp}, i \in I_j, n \in N \tag{33a}$$

$$Ts_{i,j,n} \geq Tss_{k,j,n} - UH\left(2 - wv_{i,j,n} - out_{s,k,j,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{kp}, i \in I_j, n \in N \tag{33b}$$

$$Tf_{i,j,n} \leq Tsf_{k,j,n} + UH\left(2 - wv_{i,j,n} - out_{s,k,j,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{kp}, i \in I_j, n \in N \tag{33c}$$

$$Tf_{i,j,n} \geq Tsf_{k,j,n} - UH\left(2 - wv_{i,j,n} - out_{s,k,j,n}\right), \quad \forall s \in S, k \in K_s, j \in J_k^{kp}, i \in I_j, n \in N \tag{33d}$$

All tasks should start and finish before the end of the scheduling time horizon as stated in (34a-34d).

$$Ts_{i,j,n} \leq H, \quad Tf_{i,j,n} \leq H, \qquad \forall i \in I, j \in J_i, n \in N \tag{34a}$$

$$Tss_{j,k,n} \leq H, \quad Tsf_{j,k,n} \leq H, \quad \forall j \in J, k \in K_j^{pk}, n \in N \tag{34b}$$

$$Tss_{k,j,n} \le H, \quad Tsf_{k,j,n} \le H, \quad \forall j \in J, k \in K_j^{kp}, n \in N \tag{34c}$$

$$Tos_{k,o,n} \le H, \quad Tof_{k,o,n} \le H, \quad \forall o \in O, k \in K_p, n \in N \tag{34d}$$

Intermediate due dates: Intermediate due dates for group A products, which are stored in product pools, are given by constraints (35a-35b). Orders can start unloading anytime after the vessel arrival time and finish unloading anytime before vessel departure time. The due date requirements are enforced as inequality constraints to consider demurrage by using slack variables $Tearly_o$ and $Tlate_o$ which are penalized in the objective function.

$$Tos_{k,o,n} + UH\left(1 - l_{k,o,n}\right) \ge times_o - Tearly_o, \quad \forall o \in O, k \in K_p, n \in N \tag{35a}$$

$$Tof_{k,o,n} + UH\left(1 - l_{k,o,n}\right) \le timef_o + Tlate_o, \quad \forall o \in O, k \in K_p, n \in N \tag{35b}$$

4.3.1. Valid Inequities

Valid inequalities are added in the proposed model to improve the computational efficiency of the proposed model. Constraints (36a-36b) enforce that if the material s is flowing into the tank at event point n, then the binary variable $y_{s,k,n}$ is 1 and similarly, if the material is flowing out of the tank, then the binary variable is also 1.

$$\sum_{j \in J_k^{pk}} in_{s,j,k,n} \le \sum_{j \in J_k^{pk}} y_{s,k,n}, \quad \forall k \in K, s \in S_k, n \in N \tag{36a}$$

$$\sum_{j \in J_k^{kp}} out_{s,j,k,n} \le \sum_{j \in J_k^{kp}} y_{s,k,n}, \quad \forall k \in K, s \in S_k, n \in N \tag{36b}$$

If the tank is sending or receiving the material from a production unit, then that unit is active and these requirements are represented by equations (37a-37b).

$$\sum_{k \in K_j^{pk}} in_{s,j,k,n} \le \sum_{k \in K_j^{pk}} \sum_{i \in I_j \cap I_s^p} wv_{i,j,n}, \quad \forall j \in J, s \in S_j^p, n \in N \tag{37a}$$

$$\sum_{k \in K_j^{kp}} out_{s,k,j,n} \le \sum_{k \in K_j^{kp}} \sum_{i \in I_j \cap I_s^c} wv_{i,j,n}, \quad \forall j \in J, s \in S_j^c, n \in N \tag{37b}$$

Since loading and unloading cannot happen at the same event point, if the material is unloaded at event point $n+1$ then tank is not empty at previous event point n. This is feature of model is captured by equation (38).

$$\sum_{o} l_{k,o,n+1} \leq \sum_{s \in S_k} y_{s,k,n}, \quad \forall k \in K, n \in N, n < N \tag{38}$$

If two units are consecutive without any storage tank between them, then constraint (39) impose the simultaneous operation of these units due to the continuous operation mode. However, this constraint is not imposed on parallel production units that can produce the same type of products.

$$\sum_{j' \in J_j^{seq} / J^h, i' \in I_{j'}} wv_{i',j',n} \leq \sum_{j' \in J_j^{seq} / J^h, i \in I_j} wv_{i,j,n}, \quad \forall j \in J, j \notin J^h, n \in N \tag{39}$$

Constraint (40) enforces the material balance constraint in addition to the constraint presented in equation (2c).

$$\begin{aligned} &\sum_{k \in K_j^{kp} \cap K_s} Kof_{s,k,j,n} + \sum_{j' \in J_j^{seq} \cap J_s^p} JJf_{s,j',j,n} + Uif_{s,j,n} \leq \\ &\qquad \sum_{k \in K_j^{pk} \cap K_s} Kif_{s,j,k,n} + \sum_{j' \in J_j^{seq} \cap J_s^c} JJf_{s,j,j',n} + Uof_{s,j,n}, \quad \forall j \in J, n \in N \end{aligned} \tag{40}$$

Objective Function:

The objective function (41) is used to maximize the performance and profit of total production. The refinery performance is represented by the minimization of utilization of units and tanks, all the connection between production units and tanks, start up set-ups, changeovers, and production downgrading. The blending task is significantly improved by reducing the quantities of downgraded products. Logistics and quality give-aways, under and over production, and demurrage are penalized. Profit term includes the costs of feeds and revenue of products. The penalty weights are assigned arbitrary to each term depending on its importance in schedule. The deviation from intermediate due dates is heavily penalized, quality give-aways are penalized the second most, while connection between unit and tank is least heavily penalized. It is favorable that the multipurpose product tanks store different grade products only in a certain order that is allowed by the sequence dependent switchovers constraint. The favorable switchovers are from higher grade of product to lower grade and unfavorable switchovers are from lower grade to higher grade product. Due to contamination issues, unfavorable switchovers are more heavily penalized than favorable. Similar to multipurpose tanks, there is a sequence dependent switchover restriction for multipurpose blend units. The switchover from the run mode that produces better quality product to lower quality product is least penalized than vice versa.

$$\begin{aligned} z = & \sum_{i,j\in J_i,n} c^1_{i,j} wv_{i,j,n} + \sum_{s,k\in K_s,n} c^2_k y_{s,k,n} + \sum_{k\in K_p,o,n} c^3_k l_{k,o,n} + \sum_{j,s\in S^p_j,k\in K_s,n} c^4_{j,k} in_{s,j,k,n} \\ & + \sum_{j,s\in S^c_j,k\in K_s,n} c^4_{k,j} out_{s,k,j,n} + \sum_{j\in J^h,n} c^5_j \alpha_{j,n} + \sum_{k\in K^h,n} c^6_k \beta_{k,n} + \sum_{j\in J^m,i\in I_j,i'\in I_j,i'\neq i,n} c^7_{i,i'} \chi_{i,i',j,n} \\ & + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s'} c^8_{s,s'} \eta o_{s,s',k} + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s',n} c^8_{s,s'} \eta_{s,s',k,n} + \sum_{s,k\in K_s,n} c^9_k st_{s,k,n} \\ & + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s',n} c^{10}_{k\in K^m} std_{s,s',k,n} + \sum_{k\in K^m,s\in S_k,n} c^{11}_{s,k} mh_{s,k,n} + \sum_{s\in S_b,p,n} c^{12}_{s,p} pg^l_{s,p,n} \\ & + \sum_{s\in S_b,p,n} c^{13}_{s,p} pg^u_{s,p,n} + \sum_{o\in O} c^{14}_o dg^l_o + \sum_{o\in O} c^{15}_o dg^u_o + \sum_{s\in S_f} c^{16}_s rg_s + \sum_{o\in O} c^{17}_o Tearly_o \\ & + \sum_{o\in O} c^{18}_o Tlate_o + \sum_{k,s\in S_k,n} c^{19}_s Rif_{s,k,n} + \sum_{s,j^c_s,n} c^{19}_s Uif_{s,j,n} - \sum_{s\in S_f,j^p_s,n} c^{20}_s Uof_{s,j,n} \\ & - \sum_{s\in S_b,k\in K_s,o\in O_s,n} c^{21}_s Lf_{s,k,o,n} \end{aligned} \tag{41}$$

Note that the different penalty parameters have significant effect on the computational time required to obtain an optimal solution.

4.4. Results on the Case Studies

In this section two problems based on the realistic case study presented in section 4.2.1 are solved to optimality and analyzed to show the effectiveness of the proposed model. Each case study includes different set of examples that differ in either demands, intermediate due dates, or initial hold up in the tank. All the problems are solved on a Dell Precision (Intel^{R} Xeon^{TM} with CPU 3.20 GHz and 2 GB memory) running on Windows XP using CPLEX 12.1.0/GAMS 23.2. Raw materials and finished product prices are given in Table 4.1 and the penalties parameter used in the objective function are given in Table 4.2.

Table 4.1 Price of raw materials and final products

Material	Price	Material	Price	Material	Price
ANS crude oil	25	Jet Fuel	70	Pentane	40
SJV crude oil	20	Refinery Gases	50	NC4	40
Carb Diesel	80	Isomerate	60	Alkylate	60
EPA Diesel	60	Reformate	60	FCC Gas	45
Red Dye Diesel	50	Isomax Gasoline	65	Coke	30

Table 4.2 Penalty parameters in objective function

Penalty Parameter	Value	Penalty Parameter	Value
$C^{1}_{carbnormal,diesel\,blender}$	125	$C^{11}_{s,k}$	10
$C^{1}_{EPAnormal,diesel\,blender}$	105	$C^{12}_{s,p}$	1000
$C^{1}_{\mathrm{Re}d\,dye\,normal,diesel\,blender}$	85	$C^{13}_{s,p}$	1300
C^{2}_{k}	1	C^{14}_{o}	1150
C^{3}_{k}	1	C^{15}_{o}	500
$C^{4}_{j,k}$	4 (multipurpose tank: 6)	C^{16}_{s}	1100
C^{5}_{j}	1	C^{17}_{o}	1400
C^{6}_{k}	150	C^{18}_{o}	1250
$C^{7}_{i,i'}$	40 (Unfavorable: 50)	C^{19}_{s}	$price(s)\cdot(e+q)^{-1}$
$C^{8}_{s,s'}$	60 (unfavorable: 70)	C^{20}_{s}	$price(s)\cdot e^{-1}$
C^{9}_{k}	$\left(V^{\max}_{k}\right)^{-1}$	C^{21}_{s}	$price(s)\cdot q^{-1}$
C^{10}_{k}	$\left(V^{heel}_{k}\right)^{-1}$		

Where, $e = 3\sum_{s} r_s$ and $q = 2\sum_{o,s} order_{o,s}$.

4.4.1. Case study 1

The first case study is obtained by deleting three diesel product tanks to reduce the model size and to obtain medium scale case study. The medium scale refinery has only one dedicated tank for each three diesel product and one multipurpose tank that can service CARB and EPA diesel. For this problem, there are no product tanks receiving material from more than one blender unit and no raw material tank supplying material to multiple CDUs. Thus we exclude constraints (28a) and (28b) from the model because the time sequence requirements for intermediate tanks are satisfied by other sequence constraints present in the formulation. Data for different set of examples is given in Table 4.3 and Table 4.4 and results are shown in Table 4.5 and Table 4.6. The initial hold up in Table 4.4 is stated for multipurpose tanks. Demand of group B products has to be met before the end of the scheduling time horizon. In this case study, the scheduling horizon is 72 hours (3 days). The products P1, P2, P3, and P4 correspond to Red dye diesel, EPA diesel, CARB diesel and Jet fuel, respectively. In this case study, we have included 4 product qualities requirements for blend products. The rule of thumb for the smallest number of event points needed to obtain an optimal solution is (d+1) where *d* represents the total number of diesel products in the demand orders. For example, if only CARB and EPA diesel products are required, then 3 event points should be considered first.

For the examples addressed in the case study 1, the aforementioned valid inequalities allowed us to compute medium scale scheduling problems with significantly less computational effort. The first integer solution is obtained within 15 seconds for all the examples studied. The computational performance without and with valid inequalities is presented in Table 4.5 and Table 4.6, respectively. When the valid inequalities are included in the model, the number of variables remains the same, but the number of constraints and nonzero elements increase. Valid inequalities have no effect on quality of the optimal solution; rather their effect is concentrated in significantly reducing the computational effort needed to find the optimal solution. The CPU time required to reach optimal solution is reduced by 90% when the valid inequalities are present versus when they are not included in the model. When example 3 is solved without valid inequalities, the optimal solution is not obtained even after 28 hours, whereas, when it is solved with valid inequalities, the optimal solution is obtained within 1 hour. The size of the model increases as the event point increases and time to obtain optimal solution also increases as observed for examples 4. In example 1, in order to satisfy the demand of order 2 (O2) within its delivery window, higher quality product P1 is downgraded to lower quality product P2. Multipurpose tank inventory data for example 1 is shown in Table 4.7. Product degradation is observed in optimal solution of example 3 in order to satisfy the product demand of order 2.

Table 4.3 Group A products demand order data for case study 1

Ex.	**Orders (Product type, amount(kbbl), delivery window, delivery rate(kbbl/h))**						
	O1	**O2**	**O3**	**O4**	**O5**	**O6**	**O7**
1	P3 [10,100] [58,71] 3	P2 [50,150] [10,20] 10	P1 [50,175] [28.5,46.5] 10	P4 [10,150] [40,70] 3			
2	P3 [37,100] [65,72] 10	P2 [55,150] [55,68] 10	P1 [98,175] [30,43] 10	P4 [112,150] [38,50] 10	P4 [50,100] [59.6,71.5] 10		
3	P3 [5,15] [10,23] 10	P2 [5,21] [23,30] 10	P1 [15,50] [36,48] 10	P1 [10,75] [54,65] 10	P4 [50,150] [25,38] 10	P4 [25,75] [55,72] 10	P2 [10,38] [63,72] 10
4	P3 [5,15] [10,23]	P2 [5,21] [23,30]	P1 [15,50] [36,48]	P1 [10,75] [54,65]	P4 [50,150] [25,38]	P4 [25,75] [55,72]	P2 [10,38] [63,72]

	10	10	10	10	10	10	10

Table 4.4 Group B products demands data for case study 1

Ex.	Initial holdup (product, kbbl)	Group B products demand (kbbl)								
		P5	P6	P7	P8	P9	P10	P11	P12	P13
1	P1 - 10	5	20	20	20	5	0	5	5	0
2	-	5	24	65	13	5	0	5	5	0
3	P1 - 7	5	20	15	8	5	3	5	4	9
4	P2 - 7	5	20	15	8	5	3	5	4	9

Table 4.5 Computational performance of case study 1 (without valid inequalities)

Ex.	n	0-1 Var.	Cont. Var.	Const.	Nonzero Elements	Nodes	Iterations	CPU Time (s)	Obj. Value	% gap
1	4	260	1683	4200	14841	167899	24414456	12504.70	1125.98	0.00
2	4	268	1719	4311	15245	379243	58903042	35273.88	1643.92	0.00
3	4	284	1793	4542	16024	1000000	161530640	102634.73	1234.92	3.75*
4	4	284	1794	4541	16021	1000000	204485505	121729.67	1193.73	5.01*
4	5	355	2242	5743	20702	564000	218078428	169094.88	1065.40	16.30**

*Nodes limit reached, ** out of memory

Table 4.6 Computational performance of case study 1 (with valid inequalities)

Ex.	n	0-1 Var.	Cont. Var.	Const.	Nonzero Elem	Nodes	Iterations	CPU Time (s)	Obj. Value	% gap
1	4	260	1683	4474	15675	10552	1487406	713.28	1125.98	0.00
2	4	268	1719	4585	16085	20085	2684973	1309.78	1643.92	0.00
3	4	284	1793	4816	16876	56424	7003479	3473.47	1234.24	0.00

4	4	284	1794	4815	16873	14922	1810327	983.47	1193.54	0.00
4	5	355	2242	6087	21772	78318	17482935	11616.69	1055.45	0.00

Table 4.7 Material flow in and out of multipurpose tank for case study 1, example 1

Initial Hold Up (product/amount, kbbl)	**Flow Direction**	**Event Points**	
		n1	n2
P1/10	loading	P2 (40 kbbl)	
	unloading		P2 (50 kbbl)

4.4.2. Case Study 2

In this section the proposed model is applied to Honeywell Hi-Spec refinery problem presented in section 4.2.1. We exclude constraints (28a) and (28b) because of the reasons mentioned in the previous case study. The time horizon considered in this case study is 10 days (240hrs) and there are four different quality restrictions placed on the blend products. The data for different demand orders are shown in Table 4.8 and Table 4.9 and results for these data are shown in Table 4.10. All the examples reach the first integer solution within 20 seconds. In many instances, the optimal solution is reached fast and the rest of the time is spent proving the global optimality which is a typical behavior of mixed integer programming models. In example 1, the optimal solution is obtained with product P4 sulfur limit violations and due date violation (demurrage) of demand order 1. When only 4 event points are used for example 3, the optimal solution of 3866.22 is obtained that does not fully satisfy demand order 7 (O7). However, when 5 event points are used, the optimal solution of 1562.13 is obtained that satisfies all the demand orders within their due dates. In the case of example 5 with 5 event points, the first integer solution of 286630.74 and 99.78% gap is reached within 10 seconds and the first integer solution that does not violate any demand requirements and due date restrictions is obtained within 1600 seconds with an objective value of 1741.81 and gap of 58.89 %. Example 6 obtains the first integer solution with objective value of 287161.44 and 99.91% gap in 15 seconds. The solution without any quantity, quality, and demurrage violations is reached within 450 seconds with objective value of 1728.68 and 82.08% gap. A solution without product downgrading is obtained for example 6 when 6 event points are used. As the demand order increases, the event points needed to reach best solution also increases, thus size of the model increases too.

Table 4.8 Group A products demand order data for case study 2

Orders	**Product type, amount(kbbl), delivery window, delivery rate (10 kbbl/h)**					
	Ex 1	**Ex 2**	**Ex 3**	**Ex 4**	**Ex 5**	**Ex 6**
O1	P3 [50,100] [28.5,46.5]	P3 [75,100] [110,140]	P3 [15,50] [36,54]	P3 [15,50] [36,54]	P1 [15,50] [36,54]	P1 [15,50] [24,50]
O2	P2 [50,150] [65,89]	P2 [50,100] [63,89]	P2 [32,50] [63,92]	P2 [32,50] [63,92]	P2 [32,50] [63,92]	P1 [10,50] [63.73]
O3	P1 [50,200] [105,130]	P1 [75,125] [26.5,60]	P1 [45,70] [103,119]	P1 [45,70] [103,119]	P3 [45,70] [103,119]	P2 [5,70] [85,109]
O4	P1 [50,175] [216.5,235.5]	P1 [100,175] [215,235.5]	P1 [63,98] [135,148.5]	P1 [63,98] [135,148.5]	P1 [63,98] [135,148.5]	P3 [13,98] [125,138.5]
O5	P4 [75,200] [90,120]	P4 [100,200] [90,120]	P4 [50,150] [100,135]	P4 [50,150] [100,135]	P4 [50,150] [100,135]	P4 [20,150] [100,125]
O6	P4 [50,250] [174,200]	P4 [75,250] [204,230]	P4 [15,130] [200,236.5]	P4 [15,130] [200,236.5]	P4 [15,130] [200,236.5]	P4 [25,130] [200,236.5]

O7	P2 [50,100] [166,211.5]	P2 [50,100] [166,191.5]	P2 [10,48] [166,199.5]	P2 [10,48] [166,199.5]	P2 [10,48] [166,205]	P1 [10,48] [145,175]
O8			P3 [10,35] [226,240]	P3 [10,35] [226,240]	P1 [10,35] [226,240]	P2 [10,35] [195,220]
O9						P3 [10,35] [230,240]
O10						P4 [25,150] [225,240]

Table 4.9 Group B products demands data for case study 2

Ex.	**Initial holdup (product, kbbl)**	**Group B products demand (kbbl)**								
		P5	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**
1	-	10	50	50	50	10	10	15	15	0
2	-	5	50	200	50	50	10	40	10	50
3	P2-8	7	43	172	53	40	11	44	15	30
4	-	7	43	172	53	40	11	44	15	30
5	-	7	43	172	53	40	11	44	15	30
6	P1-10	2	13	72	30	10	5	22	5	0

Table 4.10 Computational results for case study 2 (with valid inequalities)

Ex.	**Event Points**	**Var. Int./Cont. (Constraints)**	**Last Integer Solution**				**Optimal Solution**		
			Nodes/ Iterations	Obj. Value	CPU Time	Gap (%)	Nodes/ Iterations	Obj. Value	CPU Time (s)

		Nonzero Elem.			(s)				
1	4	328/1956 (5333) 18614	103384/ 9312323	34638.64	5000	0.45	249153/ 14282981	34638.64	7910.20
2	4	328/1957 (5333) 18622	20760/ 3626885	1671.86	1800	6.57	35489/ 4741152	1671.86	2449.83
3	4	336/1994 (5431) 18947	79969/ 15494045	3866.22	8300	3.64	207687/ 23440793	3866.22	12762.77
3	5	420/2490 (6872) 24466	419747/ 119257871	1562.13	100700	1.96	872647/ 133443528	1562.13	112879.86
4	4	336/1993 (5438) 18992	43412/ 8367727	1743.87	4650	6.70	59005/ 9238856	1743.87	5150.23
5	4	340/2009 (5518) 19244	22787/ 4473417	1649.75	2650	3.16	55538/ 6353433	1649.75	3787.66
5	5	425/2510 (6976) 24793	97379/ 30910982	1511.44	21500	44.30	509585/ 178283024	1511.44	136811.42*
6	5	445/2597 (7246) 25694	428485/ 160549580	1137.57	132500	38.10	916986/ 281680876	1137.57	234163.58**
6	6	534/3114 (8778) 31779	120720/ 52056975	997.25	54800	40.71	375832/ 150550479	997.25	161182.4***

* Out of memory. Optimal solution obtain with 33.13% gap

** Out of memory. Optimal solution obtain with 12.62% gap

*** Out of memory. Optimal solution obtain with 30.59% gap

4.5. Summary

In this chapter, a short-term scheduling model is developed based on continuous time representation for large-scale refineries. The model features logistics decisions such as start-up, minimum run-length, fill-draw-delay, one-flow out of blender, sequence dependent changeovers, maximum heel quantity, and downgrading of product. A set of valid inequalities are proposed that reduces the CPU resolution time by a significant factor for large scale refinery problems. The model with valid inequalities is applied to different examples and was observed that valid inequalities result in up to 90% reduction in CPU performance time compared to model without inequalities. The model is applied to two case studies to illustrate the applicability of the proposed formulations to large scale refinery operations. However, even with inclusion of the valid inequalities in the scheduling model, computational expense required to reach an optimal solution is still considerably high for the real-life oil-refinery applications. Hence, there is a need to employ different decomposition approaches such as mathematical, heuristics, or combination of both heuristics and mathematical decomposition to enable the solution of large-scale problems in a reasonable timeframe and bridge the gap between theory and industrial applicability.

Nomenclature

Indices

i	Tasks
j	Production units
k	Storage tanks
n	Event points
o	Product order
p	Properties

s	States

Sets

I_j	Tasks which can be performed in unit *j*
I_s^p	Tasks which can produce material s
I_s^c	Tasks which can consume material s
J	Production units
J_s^c	Units that consume material *s*
J_i	Units which are suitable for performing task *i*
J^h	Units that can produce all the same products as some other unit in the refinery
J^m	Units which are suitable for performing multiple tasks
J_k^{kp}	Units that consume material *s* stored in tank *k*
J_k^{pk}	Units that produce material *s* stored in tank *k*
J_s^p	Units that can produce material *s*
$J_{j'}^{seq}$	Units that follow unit j' (no storage in between)
K	Storage tanks
K^h	tanks that can store the same products as some other tank in the refinery
K_j^{kp}	Tanks that store material consumed by unit *j*
K^m	Multipurpose tanks that can store multiple materials
K_p	Tanks that can store final products
K_j^{pk}	Tanks that store material produced by unit *j*
K_s	Tanks that can store material *s*
N	Event point within the time horizon
O	Orders for products that are stored in tanks
P	Product Properties

S	States
S_b	Group A final products, produced by blenders and stored in tanks
S_f	Group B final products, products that are not stored in tanks
S_k	Materials that can be stored in tank *k*
S_i^c	Materials that can be consumed by task *i*
S_j^c	Materials that can be consumed by unit *j*
S_i^p	Materials that can be produced by task *i*
S_j^p	Materials that can be produced by unit *j*

Parameters

$D_{o,s}^{+}, D_{o,s}^{-}$	Demand limit requirement for order *o* and product *s* that is stored in tank
r_s	Demand of the final product s at the end of the time horizon
$R_{i,j}^{\min}$ / $R_{i,j}^{\max}$	Minimum/ maximum rate of material be processed by task *i* in unit *j*
RL_i	Minimum run length for task *i*
$RU_k^{\min}$ / $RU_k^{\max}$	Minimum/maximum rate of product unloading at tank *k*
$sto_{s,k}$	Amount of state *s* that is present at the beginning of the time horizon in *k*
ta_k	Fill draw delay for product tank *k*
UH	Available time horizon
$V_k^{\max}$	Maximum available storage capacity of storage tank *k*
V_k^{heel}	Maximum heel available for storage tank *k*
$yo_{s,k}$	1 if the material *s* is present at the beginning of the time horizon in k
$\rho_{s,i}^{\min}$ / $\rho_{s,i}^{\max}$	Proportion of state *s* produced/consumed by task *i*,

Variables

Binary Variables

$wv_{i,j,n}$	Assignment of task *i* in unit *j* at event point *n*
$in_{s,j,k,n}$	Assigns the material flow of *s* into storage tank *k* from unit *j* at point *n*
$l_{k,o,n}$	Assigns the starting of product flow out of product tank *k* to satisfy order *o* at event point *n*
$out_{s,k,j,n}$	Assigns the material flow of *s* out of storage tank *k* into unit *j* at point *n*
$y_{s,k,n}$	Denotes that material *s* is stored in tank k at event point n

Positive variables

$bp_{s,i,j,n}$	Amount of material s produced task *i* in unit *j* at event point *n*
$bc_{s,i,j,n}$	Amount of material s undertaking task *i* in unit *j* at event point *n*
dg_o^l	Minimum demand quantity give-away term for order *o*
dg_o^u	Maximum demand quantity give-away term for order *o*
H	Total time horizon used for production tasks
$JJf_{s,j,j',n}$	Flow of state *s* from unit *j* to consecutive unit *j'* for consumption at point *n*
$Kif_{s,j,k,n}$	Flow of material s from unit *j* to storage tank *k* event point *n*
$Kof_{s,k,j,n}$	Flow of material s from storage tank *k* to unit *j* at point *n*
$Lf_{o,k,n}$	Flow of final product for order *o* from storage tank *k* at event point *n*
$mh_{s,k,n}$	Maximum heel give-away term for product tanks
$pg_{s,p,n}^l$	Lower limit giveaway of product quality p
$pg_{s,p,n}^u$	Upper limit giveaway of quality p for product *s*
rg_s	Minimum demand quantity give-away term for Group B product *s*
$Rif_{s,k,n}$	Flow of raw material to storage tank *k* event point *n*
$st_{s,k,n}$	Amount of state *s* present in storage tank *k* at event point *n*
$std_{s,s',k,n}$	Amount of state *s* that is downgraded to state s' in storage tank *k* at event point *n*

$Tearly_o$	Early fulfillment of order *o* than required
$Tf_{i,j,n}$	Time that task *i* finishes in unit *j* at event point *n*
$Tlate_o$	Late fulfillment of order *o* than required
$Tos_{k,o,n}$	Time that material starts to flow from tank *k* for order o at event point *n*
$Tof_{k,o,n}$	Time that material finishes to flow from tank *k* for order o at event point *n*
$Ts_{i,j,n}$	Time that task *i* starts in unit *j* at event point *n*
$Tsf_{j,k,n}$	Time that material finishes to flow from unit *j* to tank *k* at event point *n*
$Tsf_{k,j,n}$	Time that material finishes to flow from tank *k* to unit *j* at event point *n*
$Tss_{j,k,n}$	Time that material starts to flow from unit *j* to storage tank *k*
$Tss_{k,j,n}$	Time that material starts to flow from tank *k* to unit *j* at event point *n*
$Uif_{s,j,n}$	Flow of raw material *s* to production unit *j* at point *n*
$Uof_{s,j,n}$	Flow of product material *s* from unit *j* at point *n*
$\alpha_{j,n}$	For unit *j*, 1 if the unit becomes active for very first time at event point *n*
$\beta_{k,n}$	For tank k, 1 if the tank becomes active for very first time at event point *n*
$\eta_{s,s',k,n}$	Continuous 0-1 variable, 1 if material in tank *k* switchover service from s at event point *n* to s' at later event point
$\eta o_{s,s',k}$	Continuous 0-1 variable, 1 if material in tank *k* switchover service from s to s'
$\chi_{i,i',j,n}$	Continuous 0-1 variable, 1 if task at unit *j* changes from *i* at event point *n* to *i'* at later event point.

Chapter 5

5. Lagrangian Decomposition Approach for Refinery Operations Scheduling

In this chapter, mathematical decomposition based on Lagrangian relaxation is proposed for the scheduling of refinery operations from crude oil processing to the blending and dispatch of finished products. A new algorithm for Lagrangian decomposition (LD) is proposed and applied to realistic large scale refinery scheduling problem to evaluate its efficiency. A novel strategy is presented to formulate restricted relaxed sub-problems based on the solution of the Lagrangian relaxed sub-problems that take into consideration the continuous process characteristic of the refinery. This new restricted algorithm provides tighter lower bound compared to classical Lagrangian decomposition approach. Furthermore, proposed heuristic rules leads to faster improvement in upper bounds. The goal of the mathematical decomposition is to produce better solutions for those integrated scheduling problems that cannot be solved in reasonable computation times. The application of the proposed algorithm results in substantial reduction in CPU solution time, duality gap, and the total number of iterations compared to classical LD.

5.1. Introduction

Lagrangian relaxation is often used for NP-hard problems where it is observed that many problems can be considered easy problems made difficult because of presence of complicating constraints. Lagrangian decomposition creates relaxed problem that is relatively easy to solve than original problem by relaxing these complicating constraints and dualizing them in the objective function with multipliers. Lagrangian relaxation provides a lower bound on the global optimal for minimization problem. Relaxed Lagrangian problems can be decomposed into easy to solve sub-problems by fixing Lagrangian multipliers. These multipliers are then typically updated iteratively using a subgradient method (Fisher, 1985) based on the upper bound. Upper bound is usually obtained using heuristics at every iteration and iterations proceeds until duality gap lies within predefined tolerance. The use of Lagrangian relaxation was popularized in the early 1970s (Cornuejols et al., 1977; Fisher, 1973, 1981; Held & Karp, 1971; Held & Karp, 1970) and this approach has wide ranging application such as scheduling (Adhya et al., 1999; Ghaddar et al., 2014; Luh & Hoitomt, 1993; Wu & Ierapetritou, 2003), planning (Graves, 1982; Gupta & Maranas, 1999; Tang & Jiang, 2009), and integrated planning and scheduling (Calfa et al., 2013; Z. Li & Ierapetritou, 2010a; Mouret et al., 2011). Lagrangian relaxation is also used to calculate lower bound in outer approximation algorithm for crude oil scheduling by Karuppiah et al. (2008) and in branch and bound approach by Holmberg and Hellstrand (1998); Holmberg and Yuan (2000).

Refinery operations involve crude oil unloading and blending, production unit operations, and finished product blending and delivery as mention in previous chapter. The model proposed in previous chapter is an integrated unified model for production unit operations and blending operations scheduling that can be applied to different type of refinery configurations. These refinery configurations differ based on blending modes. Traditionally, blending is carried out from blend component tanks, normally used for gasoline, while “rundown blending” is normally used for diesel, jet fuel, and kerosene. In rundown blending many, if not all component streams come directly from a process unit with no component storage. Many refineries are forgoing the use component tanks due to economic pressure for reducing on-site storage inventory and safety concerns for reducing inventory of volatile materials. In many instances when blend scheduling optimization problem is investigated, traditional blending operations with components storage tanks are considered. (Castillo & Mahalec, 2014a, 2014b; Cuiwen et al., 2013; Glismann & Gruhn, 2001; J. Li & Karimi, 2011; J. Li et al., 2010; Mendez et al., 2006; Zhao et al., 2008) However, scant attention has been paid to rundown blending operations optimization problem in refinery. These traditional blending scheduling models are not designed to optimize blending operations without component storage tanks. In recent years, a long term swing in demand for diesel is expected to increase by 75% from 2010 to 2040 due to higher fuel efficiency of diesel engines and preferential tax treatment of diesel "The Outlook for Energy: A View to 2040" 2013). Furthermore, demand for jet fuel is projected to grow close to 75% and increase availability of natural gas has similar long term shift away from heating oil. Refineries that can shift production to maximize diesel and jet fuel are best positioned to serve their markets. (Polasek & Mann, 2013) Compared to gasoline blending operations, blending of diesel, jet fuel, and other middle distillate products has been neglected since they are blended without component tanks. Many refineries in APAC and Europe have blend operations without component tanks and use excel with trial and error to obtain feasible solution. (Varvarezos)

Usually, the refinery operations scheduling problem has been tackled by addressing the optimization of the three sub-operations, defined in previous chapter, independently of each other. This decentralized approach for scheduling optimization is not suitable for refineries where one or more component streams is directly blended without intermediate storage. Rundown blending optimization is challenging since many process units feed into the blender and fluctuations in the incoming flow rate and property has significant impact on finished product blend qualities and quantity. Additionally, rundown header is responsible for all the material that is transferred to it from the upstream process as seen in Figure 5.1. In a blending scenario with no intermediate tanks, the number of classic degrees of freedom is limited to the component rundown flow rate or the component qualities as manipulated variables. Thus if any changes happen in the blend recipe to meet finished product properties specifications, it would have a direct impact on the production unit operations. These interdependences of the blending operations and production unit operations require an integrated approach to scheduling.

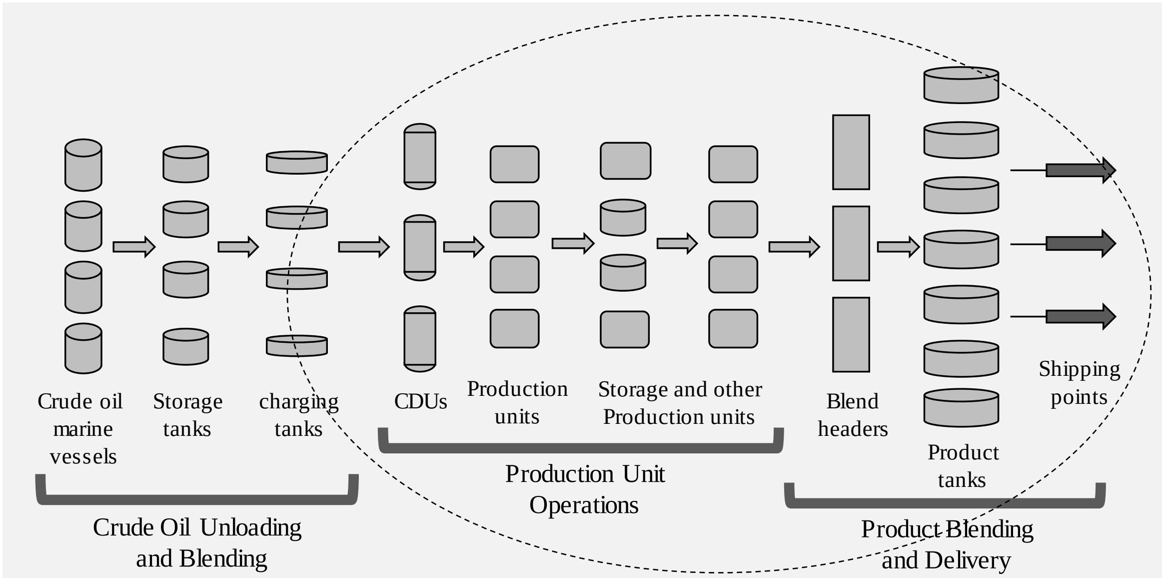

Figure 5.1 Graphic overview of a standard refinery system: includes no blend components storage tanks.

It is imperative to model an integrated production unit operations and finished product blending-delivery problem to address the issue of rundown blending. The resulting comprehensive model is complex, hard to build and solve, and has received sparse attention in the literature. Moro et al. (1998) proposed a planning model for refinery diesel production where the emphasis is on blend relations. Pinto et al. (2000) proposed a planning and scheduling model for refinery operations. They presented the formulation based on discretization of time for production and distribution scheduling and their model included the features such as sequence dependent transition cost of products within oil pipeline.(C. Luo & Rong); Chunpeng Luo and Rong (2007) developed a two decisions levels scheduling model for overall refinery. The upper level optimization model is based on discrete time formulation and it is used to determine sequencing and timing of operations modes and to decide the quantities of materials produced/consumed at each operations mode. The upper level decisions level uses aggregated tanks storage capacity whereas the lower level uses heuristics to obtain a detailed schedule. They consider multi-purpose product tanks via an iterative procedure that allows to readjust aggregated tank capacity at lower level by changing multi-purpose tank service mode and then to recalculate corresponding optimal solution at upper level. The logistics details are ensured through heuristics at the lower level.

In previous chapter we have proposed a MILP model based on continuous-time representation for the simultaneous scheduling of production unit operations and end-product blending and delivery operations. The model incorporates quantity, quality, and logistics decisions related to real-life refinery operations. These logistics details involve minimum run-length requirements, fill-draw-delay, one-flow out of blender, sequence dependent switchovers, maximum heel quantity, and downgrading of the better quality product to the lower quality. Incorporating these logistics details in blend operations can yield significant improvement in productivity and efficiency. (Kelly, 2006) We proposed a set of valid inequalities that improves the computational performance of the model significantly in section 4.3.1. However, even with valid inequalities present, the scheduling model is still computationally prohibitive. It is imperative to develop global optimization strategy for an integrated production unit and finished product blending-delivery scheduling problem to address the issue of online blend units.

The chapter is organized in the following way. Section 5.2 describes the integrated refinery operations problem. Section 5.3 presents solution strategy for the integrated problem, whereas section 5.4 defines relaxed sub-problems and presents methodology to construct novel heuristic approach to construct restricted Lagrangian sub-problems and feasible solution. Steps for obtaining tighter lower bounds are described in section 5.5 and detailed algorithm steps are presented in section 5.6. Section 5.7 provides results, and the chapter concludes with section 5.8.

5.2. Refinery Operations Scheduling

The refinery operations scheduling MILP model proposed in Chapter 4 is used in this work to serve as a benchmark for the proposed decomposition algorithm. The formulation for the refinery operations scheduling problem is based on the continuous time representation and an idea of unit-specific event-points first introduced by M. G. Ierapetritou and C. A. Floudas (1998);

M.G. Ierapetritou and C.A. Floudas (1998); Ierapetritou et al. (1999). State-task network representation introduced by Kondili et al. (1993) is used in formulating the problem. Continuous time representation is preferable because it leads to lower number of binaries and constraints compared to discrete time representation and accurate start and finish timing of tasks. Discrete formulation requires the time horizon to be divided into smaller grids to acquire an acceptable approximation of timings which leads to increasing number of binaries. (Stefansson et al., 2011) In contrasts, continuous time formulation has more complicated structure where task can start and finish at any time point. (Floudas & Lin, 2004)

The goal of the refinery operations scheduling is to maximize the production performance while minimizing the penalties subject to quantity, quality, and logistics giveaways and nonattainment. (Kelly, 2003a) The optimum values of the following variables in the system are driven by the shipment plan and accounts for the tradeoffs between costs of keeping inventory and changing run-modes: (i) sequencing of tasks for production units; (ii) each product pool that satisfies demand orders; (iii) duration of tasks at production units and duration of unloading tasks from product pools; (iv) inventory levels in component and product pools; (v) production rates for units and unloading rate for product pools; (vi) composition of material produced and consumed.

5.3. Solution Strategy

The integrated refinery scheduling model (IP) presented in previous chapter gives rise to a large scale complex MILPs problem that requires specialized solution algorithms. We apply a Lagrangian decomposition (LD) algorithm to solve the integrated scheduling problem (IP) using an iterative procedure. The LD algorithm involves relaxing complicating constraints to the objective function by introducing Lagrange multipliers to form a relaxed version of a primal problem. In LD algorithm, we obtain lower bound and upper bound of the optimal value of (IP) at each iteration.

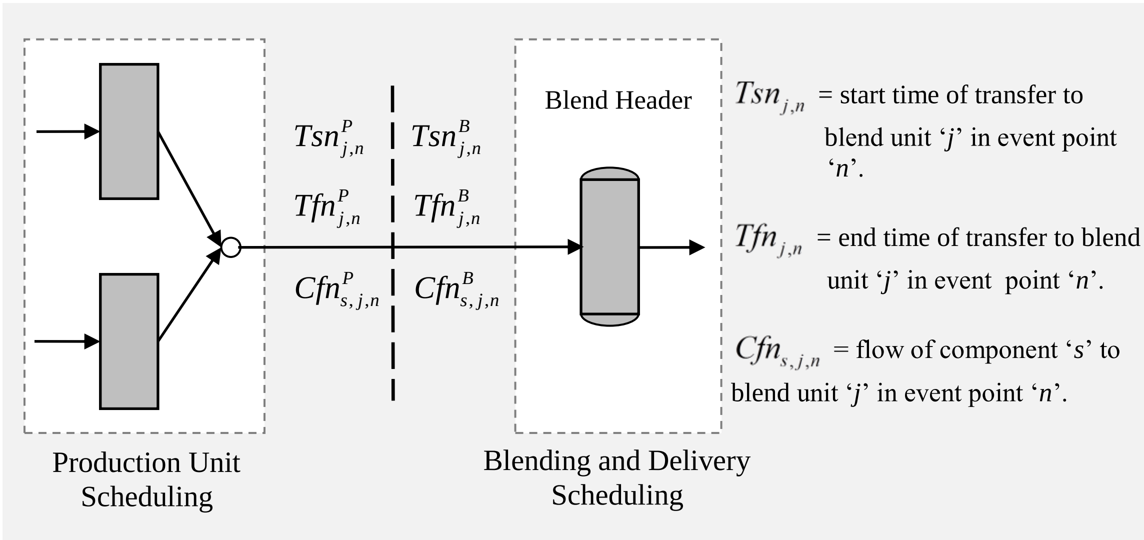

Figure 5.2 Spatial decomposition of a refinery operations network

In this work, the integrated full-scale scheduling problem is decomposed into production unit scheduling problem (PSP) and blend scheduling problem (BSP) using spatial decomposition, as shown in Figure 5.2. Here, the network is split into two independent sub-problems; furthermore, each sub-problem can be further decomposed into smaller independent sub-problems. The variables pertaining to the blend components flows amount, and start and finish times of these flows are also split alongside the connections (pipelines) in the network. Thus, two sets of linking variables, one set that belongs to the production unit scheduling sub-problem (PSP) and another set that belongs to the blend scheduling sub-problem (BSP), are present. To achieve this decomposition, coupling constraints are introduced to the integrated model (IP). Coupling constraints equate the split variables (flow amount, start and end time variables) of the blend components pipelines. Based on this spatial decomposition of the refinery structure, the main goal of the production unit scheduling sub-problem is to satisfy the demand requirement of final products that belong to set S_B and demand of the blend components required by the sub-problem. Similarly, the goal of the blend scheduling sub-problem is to satisfy the demand of the finished blend products belonging to set S_A by mixing the raw materials, supplied by (PSP), following the blending recipe and product property specifications.

5.4. Lagrangian Relaxation Framework and Sub-problems

Lagrangian relaxation provides an efficient way for obtaining lower bounds for large scale MILPs. These problems are characterized by a set of complicating constraints whose removal to objective function using Lagrangian multipliers yields a relaxed problem that can be decomposed into smaller independent sub-problems that are easier to solve. In classical LD, lower bound is obtained from the solution of the relaxed problem and upper bound is obtained by constructing a feasible solution based on a solution to the relaxed problem. Then the multipliers are updated along a sub-gradient direction.

The general framework of the classical Lagrangian decomposition (LD) algorithm is given in Figure 5.3. Here *m* is used to account for the algorithm iterations, $\left(Z^L\right)^m$ is lower bound, and $\left(Z^U\right)^m$ is upper bound. $\left(u_{j,t}^{Ts}\right)^{(m)}$, $\left(u_{j,t}^{Tf}\right)^{(m)}$, and $\left(u_{s,j,t}^{Cf}\right)^{(m)}$ are the Lagrange multipliers at the m^{th} iteration.

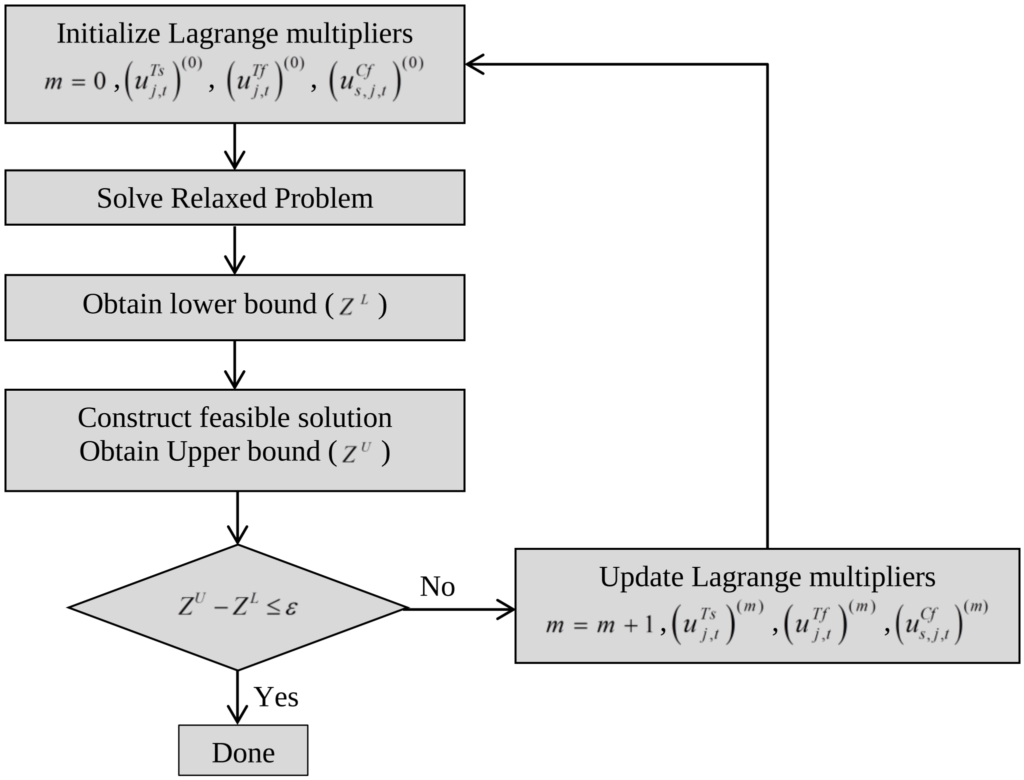

Figure 5.3 Framework of classical Lagrangian Decomposition Algorithm

In our work we develop a modified LD algorithm to suit the application to refinery continuous production processes, especially for refinery that have a blend header receiving at least one blend component stream directly from upstream processes without component storage. We formulate restricted relaxed sub-problems using the solution of Lagrangian relaxed sub-problems to obtain tighter lower and upper bounds. The details of each step in Lagrangian algorithm are given in the following subsections.

5.4.1. Relaxed Problem

The connections between production units and blend units are cut off by splitting the pipelines between them since the refinery network does not contain blend component tanks. Thus, the variables $\left\{JJf_{s,j,j',n}\right\}$, $\forall s \in S_{bc}, j \in J_s^p \cap J^{PU}, j' \in J_s^c \cap J^{BU}, n \in N$ pertaining to the blend component flow are replaced with $\left\{Cfn_{s,j',n}^{P}, Cfn_{s,j',n}^{B}\right\}$, $\forall s \in S_{bc}, j' \in J^{BU} \cap J_s^c, n \in N$. Here, J^{BU} is a set of blend units and J^{PU} is a set of units present in (PSP). Furthermore, the time sequence constraints which enforce that the start and finish time of the production units which produce blend component (s) should be same as the blend unit which consumes the blend component (s) are eliminated from the relaxed problem. In place of these sequence constraints, we introduce variables $\left\{Tsn_{j,n}, Tfn_{j,n}\right\}, \ \forall j \in J^{BU}, n \in N$ for start and finish time of flow from production unit scheduling problem to a blend unit b .

The variables pertaining to the blend component flow and start/end times of flow for all split connections between production unit operations and blend units are duplicated. The model has two sets of linking variables, one set that belongs to the production unit scheduling sub-problem $\left(Cfn^{P}_{s,j,n}, Tsn^{P}_{j,n}, Tfn^{P}_{j,n}\right)$ and another set that belongs to the blend scheduling sub-problem $\left(Cfn^{B}_{s,j,n}, Tsn^{B}_{j,n}, Tfn^{B}_{j,n}\right)$. All other variables in model (IP) are called non-linking variables since they are separate for both sub-problems.

Coupling constraints (1)-(3) are introduced to equate the split variables (flow amount, start and end time variables) for every event-point n .

$$Tsn^{P}_{j,n} = Tsn^{B}_{j,n}, \quad \forall j \in J^{BU}, n \in N \tag{1}$$

$$Tfn^{P}_{j,n} = Tfn^{B}_{j,n}, \quad \forall j \in J^{BU}, n \in N \tag{2}$$

$$Cfn^{P}_{s,j,n} = Cfn^{B}_{s,j,n}, \quad \forall j \in J^{BU}, s \in S^{c}_{j}, n \in N \tag{3}$$

To obtain the Lagrangian relaxation of the original problem, the complicating constraints (1-3) are relaxed using Lagrange multipliers (u) and considered in the objective function as shown in equation (4).

$$L^{n}\left(u\right) = z + \sum_{j \in J^{BU}, n} u^{Tsn}_{j,n}\left(Tsn^{B}_{j,n} - Tsn^{P}_{j,n}\right) + \sum_{j \in J^{BU}, n} u^{Tfn}_{j,n}\left(Tfn^{B}_{j,n} - Tfn^{P}_{j,n}\right) + \sum_{j \in J^{BU}, s \in S^{c}_{j}, n} u^{Cfn}_{s,j,n}\left(Cfn^{B}_{s,j,n} - Cfn^{P}_{s,j,n}\right) \tag{4}$$

The objective function (4) of the relaxed problem is decomposable into smaller sub-problems corresponding to the production unit operations and the finished product blending and delivery operations, which are relatively easier to solve. The model (P1-n) includes objective function (L^{Bn}) and all the constraints and variables pertaining to the finished product blending and delivery operations, whereas, the model (P2-n) includes objective function (L^{Pn}) and all the equations and variables pertaining to the production unit operations.

$$\text{minimize}\left\{ L^{Bn}\left(u\right) = z + \sum_{j\in J^{BU},n} u^{Tsn}_{j,n} Tsn^{B}_{j,n} + \sum_{j\in J^{BU},n} u^{Tfn}_{j,n} Tfn^{B}_{j,n} + \sum_{j\in J^{BU},s\in S^{c}_{j},n} u^{Cfn}_{s,j,n} Cfn^{B}_{s,j,n} \right\}$$

s.t. constraints corresponding to finished product blending and delivery operations (P1-n)

$$\text{minimize}\left\{ L^{Pn}\left(u\right) = z - \sum_{j\in J^{BU},n} u^{Tsn}_{j,n} Tsn^{P}_{j,n} - \sum_{j\in J^{BU},n} u^{Tfn}_{j,n} Tfn^{P}_{j,n} - \sum_{j\in J^{BU},s\in S^{c}_{j},n} u^{Cfn}_{s,j,n} Cfn^{P}_{s,j,n} \right\}$$

s.t. constraints corresponding to production units scheduling operations (P2-n)

The models (P1-n) and (P2-n) are independent and can be solved in parallel. Their solutions will provide a feasible solution to the original problem when the coupling constraints (1-3) are satisfied for all event-points. Figure 5.4 shows a typical solution obtained by solving (P1-n) and (P2-n) for problem that has one blend unit, say *b1*. Gantt chart shows the flow amount and start/end times of one of the blend component ($s1$) for blend unit *b1*. As you can see, blend components are being supplied from production units to the blend header and consumed by the blend header at the same time. However, if one were to look at Gantt chart closely, the event-points for the component streams are not the same. The complicating constraints (1)-(3) are not satisfied for event-points $n=2$ and $n=3$.

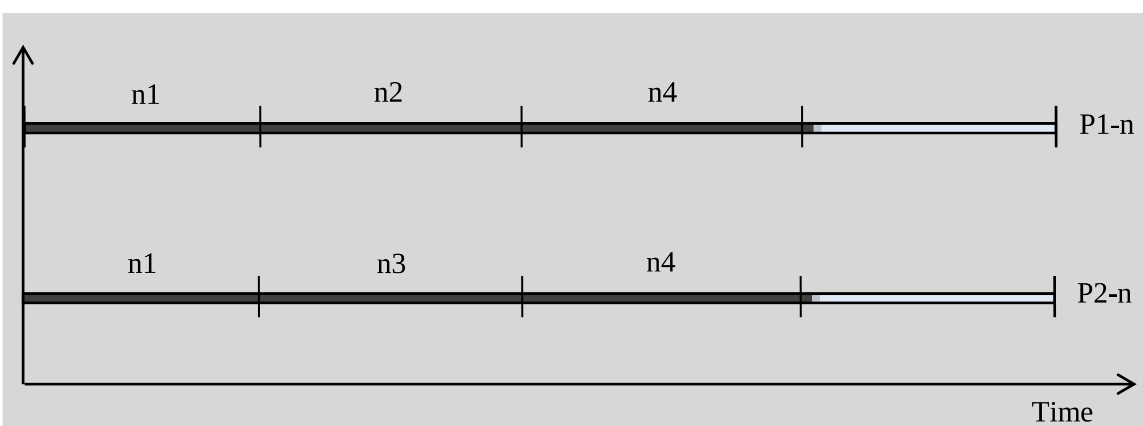

Figure 5.4 Gantt chart for a blender receiving component *s*.

This kind of situation arises frequently when comparing solutions of (P1-n) and (P2-n) across event points due to the nature of unit specific event points and continuous time representation used in the model. To meditate this kind effect, we introduce a time-point t that has one-to-one correspondence with an active event-point n and cardinality of set T is equal to the cardinality of set N. To eliminate the occurrence of non-active time-points before active time-points, additional constraints are included to enforce that active time-points always occur before non-active time-points.

To compare material flow between production operations and blend operations at each time points, a new set of variables $\left(Cf_{s,j,t}, Ts_{j,t}, Tf_{j,t}\right)$ are introduced. The coupling constraints presented in equation (1)-(3) are replaced by (1b)-(3b).

$$Ts_{j,t}^{P} = Ts_{j,t}^{B}, \quad \forall j \in J^{BU}, t \in T \tag{1b}$$

$$Tf_{j,t}^{P} = Tf_{j,t}^{B}, \quad \forall j \in J^{BU}, t \in T \tag{2b}$$

$$Cf_{s,j,t}^{P} = Cf_{s,j,t}^{B}, \quad \forall j \in J^{BU}, s \in S_j^c, t \in T \tag{3b}$$

The Lagrangian relaxation objective function $L^n(u)$ given in equation (4) is updated to $L(u)$ given in equation (4b),

$$L(u) = z + \sum_{j \in J^{BU}, t} u_{j,t}^{Ts}\left(Ts_{j,t}^{B} - Ts_{j,t}^{P}\right) + \sum_{j \in J^{BU}, t} u_{j,t}^{Tf}\left(Tf_{j,t}^{B} - Tf_{j,t}^{P}\right) + \sum_{j \in J^{BU}, s \in S_j^c, t} u_{s,j,t}^{Cf}\left(Cf_{s,j,t}^{B} - Cf_{s,j,t}^{P}\right) \tag{4b}$$

Where the variables $\left(Cfn_{s,j,n}, Tsn_{j,n}, Tfn_{j,n}\right)$ are replaced with variables $\left(Cf_{s,j,t}, Ts_{j,t}, Tf_{j,t}\right)$.

The relaxed sub-problems (P1-t) and (P2-t) are obtained by decomposing $L(u)$.

$$\text{minimize}\left\{L^B(u) = z + \sum_{j \in J^{BU}, t} u_{j,t}^{Ts} Ts_{j,t}^{B} + \sum_{j \in J^{BU}, t} u_{j,t}^{Tf} Tf_{j,t}^{B} + \sum_{j \in J^{BU}, s \in S_j^c, t} u_{s,j,t}^{Cf} Cf_{s,j,t}^{B}\right\}$$

s.t. constraints corresponding to finished product blending and delivery operations (P1-t)

$$\text{minimize}\left\{L^P(u) = z - \sum_{j\in J^{BU},t} u^{Ts}_{j,t} Ts^P_{j,t} - \sum_{j\in J^{BU},t} u^{Tf}_{j,t} Tf^P_{j,t} - \sum_{j\in J^{BU},s\in S^c_j,t} u^{Cf}_{s,j,t} Cf^P_{s,j,t}\right\}$$

s.t. constraints corresponding to production units scheduling operations (P2-t)

5.4.1.1. Relaxed Blend Scheduling Problem

Relaxed blend scheduling problem includes all the constraints and variables of the relaxed sub-problem (P1-t) and in addition to those constraints and variables; we introduce a new continuous 0-1 variable $x^B_{j,n,t}$ to connect the time-points t and event-points n.

$$x^B_{j,n,t} \le \sum_{i\in I_j} wv_{i,j,n}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{5}$$

$$x^B_{j,n,t} \le 1 - \sum_{t'<t, t'\le n} x^B_{j,n,t'}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{6}$$

$$x^B_{j,n,t} \le 1 - \sum_{n'<n, n'\ge t} x^B_{j,n',t}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{7}$$

$$x^B_{j,n,t} \ge \sum_{i\in I_j} wv_{i,j,n} - \sum_{t'<t, t'\le n} x^B_{j,n,t'} - \sum_{n'<n, n'\ge t} x^B_{j,n,t'}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{8}$$

$$\sum_{n\ge t'} x^B_{j,n,t'} \le \sum_{n\ge t} x^B_{j,n,t}, \quad \forall j \in J^{BU}, t \in T, t' \in T, t' > t \tag{9}$$

$$x^B_{j,n,t} + x^B_{j,n',t'} \le 1, \quad \forall j \in J^{BU}, n \in N, n' \in N, n < n', t \in T, t' \in T, t' < t, n \ge t, n' \ge t, n' \ge t', n \ge t' \tag{10}$$

$$x^B_{j,n,t} = 0, \quad \forall j \in J^{BU}, n \in N, t \in T, n < t \tag{11}$$

Constraint (5) states that variable $x^B_{j,n,t}$ is non-zero at time-point *t* only if a blend unit j is active at event-point *n*. Here $wv_{i.j.n}$ is a binary assignment variable, 1 if task *i* is active in unit *j* at event-point *n*, and 0 otherwise. Constraints (6) and (7) enforce that each blend unit *j* has one-to-one correspondence between a time-point and an event-point. Equation (8) states that if the blend unit is active at event-point *n* then it must correspond to some time-point *t*. Constraint (9) restrict that time-point $t' < t$ should be assign an event-point before time-point *t*. Similar to constraint (10),

$n' < n$ should be assigned to a time-point $t' < t$ before assigning *n* to *t*. Since, we want to restrict inactive time-points ($x^B_{j,n,t} = 0$) to happen after active time-points ($x^B_{j,n,t} = 1$), we add equation (11). The decoupled flow and time variables at each time-point are defined as follow:

$$Ts^B_{j,t} \le Ts_{i,j,n} + UH\left(2 - wv_{i,j,n} - x^B_{j,n,t}\right), \quad \forall j \in J^{BU}, i \in I_j, n \in N, t \in T, n \ge t \tag{12a}$$

$$Ts^B_{j,t} \ge Ts_{i,j,n} - UH\left(2 - wv_{i,j,n} - x^B_{j,n,t}\right), \quad \forall j \in J^{BU}, i \in I_j, n \in N, t \in T, n \ge t \tag{12b}$$

$$Tf^B_{j,t} \le Tf_{i,j,n} + UH\left(2 - wv_{i,j,n} - x^B_{j,n,t}\right), \quad \forall j \in J^{BU}, i \in I_j, n \in N, t \in T, n \ge t \tag{12c}$$

$$Tf^B_{j,t} \ge Tf_{i,j,n} - UH\left(2 - wv_{i,j,n} - x^B_{j,n,t}\right), \quad \forall j \in J^{BU}, i \in I_j, n \in N, t \in T, n \ge t \tag{12d}$$

$$Cf^B_{s,j,t} \le Uif_{s,j,n} + M\left(1 - x^B_{j,n,t}\right), \quad \forall j \in J^{BU}, s \in S^c_j, n \in N, t \in T, n \ge t \tag{13a}$$

$$Cf^B_{s,j,t} \ge Uif_{s,j,n} - M\left(1 - x^B_{j,n,t}\right), \quad \forall j \in J^{BU}, s \in S^c_j, n \in N, t \in T, n \ge t \tag{13b}$$

$$Ts^B_{j,t} \le UH \sum_{n \ge t} x^B_{j,n,t}, \quad \forall j \in J^{BU}, t \in T \tag{14a}$$

$$Tf^B_{j,t} \le UH \sum_{n \ge t} x^B_{j,n,t}, \quad \forall j \in J^{BU}, t \in T \tag{14b}$$

$$Cf^B_{s,j,t} \le M \sum_{n \ge t} x^B_{j,n,t}, \quad \forall j \in J^{BU}, s \in S^c_j, t \in T \tag{15}$$

Constraints (12a)-(12d) assign start and end times to time-point *t* using the corresponding event-point *n*. Similarly, blend components flow amount is assigned to time-point *t* using equation (13a) and (13b), where *M* is a big M term. If a time-point is not active then the variables pertaining to blend components flows are equal to zero as defined by equations (14a), (14b) and (15). If refinery under study contains multiple blend units, then the relaxed blend scheduling problem may be decomposed into multiple independent blend scheduling problems. Each of these independent blend scheduling problems must blend different components, more specifically, any two independent scheduling problems must not share a common blend component.

$$\sum_{n \ge t} x^B{}_{j,n,t} \le 1, \quad \forall j \in J^{BU}, t \in T \tag{16}$$

$$\sum_{t \le n} x^B_{j,n,t} = \sum_{i \in I_j} wv_{i,j,n}, \quad \forall j \in J^{BU}, n \in N \tag{17}$$

To improve the computational performance of the decomposed problems, valid inequalities (16) and (17) are added to the blend scheduling models. Inequality (16) states that for a given time-point *t*, continuous 0-1 variable $x^B_{j,n,t}$ can only correspond to one event-point and equality (17) states that if blend unit is active at given event-point *n* then there must be a time-point *t* corresponding to *n*.

5.4.1.2. Relaxed Production Unit Scheduling Problem

Continuous 0-1 variables $x^P_{j,n,t}$ and $a_{j,n}$ are introduced to define split variables belonging to production unit scheduling problem in terms of time-point *t* instead of event-point *n*. To this goal, we first introduce equations (18a) and (18b).

$$a_{j,n} \le \sum_{j' \in J^{PU}, i \in I_{j'}, S^c_j \cap S^p_i \ne \emptyset} wv_{i,j',n}, \quad \forall j \in J^{BU}, n \in N \tag{18a}$$

$$\sum_{i \in I_{j'}, S^c_j \cap S^p_i \ne \emptyset,} wv_{i,j',n} \le a_{j,n}, \quad \forall j \in J^{BU}, j' \in J^{PU}, j \in J^{seq}_{j'}, n \in N \tag{18b}$$

Binary variable $a_{j,n}$ takes the value of 1 if at least one production task produces a blend component at event-point *n* that can be consumed by blend unit *j* as stated by constraint (18a). Equation (18b) restricts $a_{j,n}$ to 0 if production operations are not producing blend components at event- point *n* for blend unit *j*.

Similarly to blend scheduling problem, production unit scheduling problem has equations (19)-(25) to define 0-1 continuous variable $x^P_{j,n,t}$.

$$x^P_{j,n,t} \le a_{j,n}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{19}$$

$$x^P_{j,n,t} \le 1 - \sum_{t' < t, t' \le n} x^P_{j,n,t'}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{20}$$

$$x^P_{j,n,t} \le 1 - \sum_{n' < n, n' \ge t} x^P_{j,n',t}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \ge t \tag{21}$$

$$x^P_{j,n,t} \geq a_{j,n} - \sum_{t'<t,t'\leq n} x^P_{j,n,t'} - \sum_{n'<n,n'\geq t} x^P_{j,n,t'}, \quad \forall j \in J^{BU}, n \in N, t \in T, n \geq t \tag{22}$$

$$\sum_{n\geq t'} x^P_{j,n,t'} \leq \sum_{n\geq t} x^P_{j,n,t}, \quad \forall j \in J^{BU}, t \in T, t' \in T, t' > t \tag{23}$$

$$x^P_{j,n,t} + x^P_{j,n',t'} \leq 1, \quad \forall j \in J^{BU}, n \in N, n' \in N, n < n', t \in T, t' \in T, t' < t, n \geq t, n' \geq t, n' \geq t', n \geq t' \tag{24}$$

$$x^P_{j,n,t} = 0, \quad \forall j \in J^{BU}, n \in N, t \in T, n < t \tag{25}$$

Constraint (19) states that variable $x^P_{j,n,t}$ is non-zero at time-point *t* only if there is at least one production unit operational at event-point *n* that can produce a blend component necessary for blend unit *j*. Constraints (20) and (21) enforce that there is one to one relationship between a time-point and an event-point for a every connections between production unit scheduling problem and blend unit *j*. Equation (22) ensures that if there is a blend component production happening event-point *n* then it must correspond to some time-point *t*. Constraint (23)-(25) are same as constraints (9)-(11) present in the blend scheduling problem.

$$\sum_{j'\in J^{PU}\cap J^P_s, i\in I_{j'}\cap I^P_s} wv_{i,j',n} \geq a_{j,n}, \quad \forall j \in J^{BU}, s \in S^c_j, n \in N \tag{26}$$

Since there are no blend component tanks present, blend unit must receive all blend components from production scheduling problem at the same time for inline continuous blending. Constraint (26) requires that if at least one blend component is produced at event-point *n* for a blend unit *j* then all other blend components for the blend unit must be produced at event-point *n*. This requirement is valid because in a typical refinery, a blend unit produces multiple products by mixing a fixed set of blend components using different blend recipes. That is, $\forall j \in J^{BU}, I \in I_j$, the blend components that can be consumed by different tasks at blend unit *j* gives us $S^c_j = S^c_i$.

The goals of the production unit scheduling problem is to produce hydrocarbon products (S_B) that are not stored in tanks but are supplied straight to the market and to produce blend components that are needed by blend units to produce finished blend products. Since there are no tanks available to store blend components, these products (blend components) are supplied straight to the market (here market is blend unit). Thus, the product set (S_B) includes the blend components and other hydrocarbon products that will be used as feedstock in chemical industry. Constraints relating the split variables (flow, start and end times) of the production scheduling problem to time-points are given below.

$$Ts^{P}_{j,t} \le UH \sum_{n \ge t} x^{P}_{j,n,t}, \quad \forall j \in J^{BU}, t \in T \tag{27a}$$

$$Tf^{P}_{j,t} \le UH \sum_{n \ge t} x^{P}_{j,n,t}, \quad \forall j \in J^{BU}, t \in T \tag{27b}$$

$$Ts^{P}_{j',t} \le Ts_{i,j,n} + UH\left(2 - wv_{i,j,n} - x^{P}_{j',n,t}\right), \\ \forall j \in J^{PU}, j' \in J^{BU}, j' \in J^{seq}_{j}, i \in I_j, n \in N, t \in T, n \ge t, S^{p}_{i} \cap S^{c}_{j'} \ne \varnothing \tag{28a}$$

$$Ts^{P}_{j',t} \ge Ts_{i,j,n} - UH\left(2 - wv_{i,j,n} - x^{P}_{j',n,t}\right), \\ \forall j \in J^{PU}, j' \in J^{BU}, j' \in J^{seq}_{j}, i \in I_j, n \in N, t \in T, n \ge t, S^{p}_{i} \cap S^{c}_{j'} \ne \varnothing \tag{28b}$$

$$Tf^{P}_{j',t} \le Tf_{i,j,n} + UH\left(2 - wv_{i,j,n} - x^{P}_{j',n,t}\right), \\ \forall j \in J^{PU}, j' \in J^{BU}, j' \in J^{seq}_{j}, i \in I_j, n \in N, t \in T, n \ge t, S^{p}_{i} \cap S^{c}_{j'} \ne \varnothing \tag{28c}$$

$$Tf^{P}_{j',t} \ge Tf_{i,j,n} - UH\left(2 - wv_{i,j,n} - x^{P}_{j',n,t}\right), \\ \forall j \in J^{PU}, j' \in J^{BU}, j' \in J^{seq}_{j}, i \in I_j, n \in N, t \in T, n \ge t, S^{p}_{i} \cap S^{c}_{j'} \ne \varnothing \tag{28d}$$

Constraints (27a) and (27b) states that start and end times are zero if material is not flowing to a blend unit *j'* at time-point *t*. If production unit operations are producing blend components at time-point *t*, then equations (28a)-(28d) are used to assign start and end times to blend component flow to a blend unit *j'*.

All production and blend units in the refinery are continuous processes and since there is no storage between the production unit scheduling operations and the blend units, material produced by the production unit scheduling problem is directly consumed. Thus production capacity of blend units and blend recipes of each blend component should also be taken into consideration when the components are supplied by production unit operations. Parameters $RB^{\max}_{j}$ and $RB^{\min}_{j}$ are maximum and minimum blend rate of a blend unit *j*, respectively and are calculated using equations (29a) and (29b). The maximum and minimum blend recipe of component *s* in blend unit *j*, $\rho b_{s,j}^{\max}$ and $\rho b_{s,j}^{\min}$ respectively, are determined from Equations (30a) and (30b).

$$RB_j^{\max} = \max_{i \in I_j}\left(R_{i,j}^{\max}\right), \quad \forall j \in J^{BU} \tag{29a}$$

$$RB_j^{\min} = \min_{i \in I_j}\left(R_{i,j}^{\min}\right), \quad \forall j \in J^{BU} \tag{29b}$$

$$\rho b_{s,j}^{\max} = \max_{i \in I_j}\left(\rho_{s,i}^{\max}\right), \quad \forall j \in J^{BU} \tag{30a}$$

$$\rho b_{s,j}^{\min} = \min_{i \in I_j}\left(\rho_{s,i}^{\min}\right), \quad \forall j \in J^{BU} \tag{30b}$$

The blend component production is set to zero by equation (31a) if there is no production happening at time-point *t* for blend component *s*. Constraints (31b) and (31c) determines the individual component flow amount from the production unit scheduling problem to a blend unit *j* at time-point *t*.

$$Cf_{s,j,t}^{P} \le \rho b_{s,j}^{\max} RB_j^{\max} UH \sum_{n \ge t} x_{j,n,t}^{P}, \quad \forall j \in J^{BU}, s \in S_j^c, t \in T \tag{31a}$$

$$Cf_{s,j,t}^{P} \le \sum_{j' \in J_s^P} Uof_{s,j',n} + \rho b_{s,j}^{\max} RB_j^{\max} UH\left(1 - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, s \in S_j^c, n \in N, t \in T, n \ge t \tag{31b}$$

$$Cf_{s,j,t}^{P} \ge \sum_{j' \in j_s^P} Uof_{s,j',n} - \rho b_{s,j}^{\max} RB_j^{\max} UH\left(1 - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, s \in S_j^c, n \in N, t \in T, n \ge t \tag{31c}$$

$$\sum_{s \in S_j^c} Cf_{s,j,t}^{P} \le RB_j^{\max}\left(Tf_{j,t}^{P} - Ts_{j,t}^{P}\right) + RB_j^{\max} UH\left(1 - \sum_{n \ge t} x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, t \in T \tag{32a}$$

$$\sum_{s \in S_j^c} Cf_{s,j,t}^{P} \ge RB_j^{\min}\left(Tf_{j,t}^{P} - Ts_{j,t}^{P}\right) - RB_j^{\min} UH\left(1 - \sum_{n \ge t} x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, t \in T \tag{32b}$$

$$Cf_{s,j,t}^{P} \le \rho b_{s,j}^{\max} RB_j^{\max}\left(Tf_{j,t}^{P} - Ts_{j,t}^{P}\right) + \rho b_{s,j}^{\max} RB_j^{\max} UH\left(1 - \sum_{n \ge t} x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, s \in S_j^c, t \in T \tag{32c}$$

$$Cf_{s,j,t}^{P} \ge \rho b_{s,j}^{\min} RB_j^{\min}\left(Tf_{j,t}^{P} - Ts_{j,t}^{P}\right) - \rho b_{s,j}^{\min} RB_j^{\min} UH\left(1 - \sum_{n \ge t} x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, s \in S_j^c, t \in T \tag{32d}$$

Constraints (32a) and (32b) enforce that total amount of material supplied to a blend unit *j* at time-point *t* is bounded by maximum and minimum production rate of corresponding blend unit and production of each blend component is bounded by equations (32c) and (32d).

Similar to valid inequalities (16) and (17) included in blend scheduling sub-problem, we introduce constraints (33) and (34) to production unit scheduling problem. Inequality (33) states that for a given time-point *t*, continuous 0-1 variable $x^P_{j,n,t}$ can only correspond to one event-point. If at event-point *n* production unit operations are producing materials for blend unit *j*, then there must be at least one time-point *t* corresponding to *n* as stated by equality (34).

$$\sum_{n \geq t} x^P{}_{j,n,t} \leq 1, \quad \forall j \in J^{BU}, t \in T \tag{33}$$

$$\sum_{t \leq n} x^P_{j,n,t} = a_{j,n}, \quad \forall j \in J^{BU}, n \in N \tag{34}$$

5.4.2. Construction of a Feasible Schedule

Traditionally an upper bound for the Lagrangian decomposition is obtained by fixing certain binary variables of the original full-scale integrated MILP problem (IP) based upon the solution of the relaxed sub-problems. However the solution derived by solving relaxed sub-problems ((P1-t) and (P2-t)) are usually infeasible in each iteration because the complicating constraints are are violated. Therefore, a heuristics procedure or some other methods for generating feasible solution is necessary. In this work, feasible solution at each iteration is obtained by fixing some of the binary variables to 1 and to decide which binaries to fix, we propose using solution of the relaxed sub-problems to build new restricted relaxed sub-problems. These restricted relaxed sub-problems' solution will always provide feasible fullspace solution unlike relaxed sub-problems.

In proposed decomposition strategy, complicated variables are expressed in terms of event-point t and not event-point n, thus the schedule obtained solving relaxed problem (P1-t) and (P2-t) would lead to an infeasible solution of the (IP). In most instances the infeasibility in full-scale model occurs because the fundamental operational rule of the refinery is violated because material balance constraints are not satisfied for blend units or production units that produce blend components. The refinery system under study doesn't contain blend component tanks thus blend operations and corresponding production unit operations must operate concurrently to satisfy mass balance constraints of rundown streams. Relaxed sub-problems (P1-t) and (P2-t) defined in previous section are solved independent of each other and their solutions are compared at time-point t. Thus at some iteration m, the production operations scheduling model (P2-t) may produce a solution where blend components are supplied to a blend unit j during event-points $n=2$ and $n=4$, that is $x_{j,2,1}^{P}=1$ and $x_{j,4,2}^{P}=1$, while relaxed blend scheduling model (P1-t) solution has the blend unit j is receiving components during event-points $n=1$ and $n=3$, that is $x_{j,1,1}^{B}=1$ and $x_{j,3,2}^{B}=1$. Hence at if the binary variables $\{wv_{i,j,n}\}\ \forall i,j,n$ in the original integrated model (IP) are fixed based upon the solution of relaxed sub-problems, it would lead to an infeasible solution. Infeasibility would occur because in the refinery, for two sequential units without storage any between them must operate at the same event-point. Hence sub-problems must provide a solution where a blend unit $j \in J^{BU}$ is receiving and blending components at the same event-points. That is, both $x_{j,n,t}^{P}$ and $x_{j,n,t}^{B}$ have to be nonzero at event point n to provide feasible solution of full space problem.

Before the steps to build restricted relaxed sub-problems are presented, we first state assumptions behind proposed method to derive restricted relaxed sub-problems.

We assume that no parallel blend units are present in the refinery and no two blend units share a particular blend component. These assumptions are valid because, in this work, the refinery does not have any blend component tanks present. The mathematical representation of these assumptions is given below as:

$$S_j^c \cap S_{j'}^c = \varnothing, K_j^{pk} \cap K_{j'}^{pk} = \varnothing \quad \forall j \in J^{BU}, j' \in J^{BU}, j \neq j'$$

Thus, blend scheduling problems (BSP) can be decomposed into sub-problems containing only one blend unit. Here each (BSP) sub-problem defined as $BSP^{j \in J^{BC}}$ has its own set of blend components, materials, and tanks corresponding a blend unit $j \in J^{BC}$.

Now we analyze the structure of the (PSP), particular for a refinery that has more than one blend unit and the (BSP) can be decomposed into independent sub-problems. We check if the production units that supply components to different (BSP) sub-problems are connected or not. Production units are said to be "connected" if they all have to be operational at the same time which corresponds to being active at same event-points so that the continuous process operational rule is not violated for the (PSP). We assume in our work that all the production units which produce blend components for different blend units are connected. Based upon this assumption, all independent sub-problems of the (BSP) must also be active at the same event-point to obtain a feasible solution to the integrated model (IP).

A two steps approach is developed to determine an upper bound for the algorithm which is obtained from a feasible solution to the original full-scale model. In step 1 a restricted relaxed problem is solved to obtain a solution that has production units active at the same event-points as the blend units. Using the solution of step 1, in step 2 a feasible solution to (IP) is realized.

5.4.2.1. Restricted Relaxed Problem

Upon solving relaxed sub-problems presented in section 5.4.1, we obtain schedule for blending operations and production unit operations, respectively. To build a restricted relaxed problem using the optimal solution of relaxed problem, the following steps are proposed:

Step 1: From the optimal relaxed sub-problems solution, determine over the scheduling horizon which tasks, units, and storage tanks are active for at least one event-point.

Step 2: Pick a primary sub-problem from $\{PSP\} \cup \{BSP^{j}, \forall j \in J^{BU}\}$ by analyzing solution of the corresponding relaxed sub-problems. Determine active event-points during which the primary sub-problem is either supplying or receiving blend components.

Step 3: Obtain restricted relaxed problem by fixing some of the binary variables in corresponding relaxed problem to 1 at active event-points.

Step 1: Determine parameters from an optimal solution of relaxed sub-problems. Using equations (35a)-(35d), we can calculate active task for units ($wv^a_{i,j}$), the flow in and out of tanks ($in^a_{s,j,k}$ and $out^a_{s,k,j}$), and the changeovers from task i to i' at unit j, respectively over the scheduling horizon. Parameter $\bar{\chi}^a_{i,i',j}$ is calculated using equation (35e) which gives the total number of time a task i is change to a task i' at unit j.

$$wv^a_{i,j} = \begin{cases} 1 & if \sum_n wv_{i,j,n} > 0 \\ 0 & if \sum_n wv_{i,j,n} = 0 \end{cases} \quad \forall j \in J, i \in I_j \tag{35a}$$

$$in^a_{s,j,k} = \begin{cases} 1 & if \sum_n in_{s,j,k.n} > 0 \\ 0 & if \sum_n in_{s,j,k,n} = 0 \end{cases} \quad \forall s \in S, k \in K_s, j \in J_k^{pk} \tag{35b}$$

$$out^a_{s,k,j} = \begin{cases} 1 & if \sum_n out_{s,k,j,n} > 0 \\ 0 & if \sum_n out_{s,k,j,n} = 0 \end{cases} \quad \forall s \in S, k \in K_s, j \in J_k^{kp} \tag{35c}$$

$$\chi^a_{i,i',j} = \begin{cases} 1 & if \sum_n \chi_{i,i',j,n} > 0 \\ 0 & if \sum_n \chi_{i,i',j,n} = 0 \end{cases} \quad \forall j \in J, i \in I_j, i' \in I_j \tag{35d}$$

$$\bar{\chi}^a_{i,i',j} = \sum_n \chi_{i,i',j,n} \quad \forall j \in J, i \in I_j, i' \in I_j \tag{35e}$$

Step 2: From the solution of relaxed sub-problems, the total number of active time-points for each Lagrangian decomposed problem are calculated as:

$$\mathrm{Tat}_j = \sum_{t,n \geq t} x_{j,n,t}, \quad \forall j \in J^{BU}$$

Since we assume that the production units which produce blend components for different blend units are connected, for (PSP),

$$\mathrm{Tat}^P = \mathrm{Tat}^P_j = \mathrm{Tat}^P_{j'}, \quad \forall j \in J^{BU}, j' \in J^{BU}, j \neq j'$$

The total numbers of active-points for each sub-problem BSP^{j} containing blend unit *j* are given as Tat_{j}^{B}. The maximum number of active time-points is determined by finding maximum among (BSP) and (PSP) as:

$$Mat = \max\left[\left\{\mathrm{Tat}^{P}\right\} \cup \left\{\mathrm{Tat}_{j}^{B}, \forall j \in J^{BU}\right\}\right]$$

If there is more than one sub-problem that has the total active time-points equal to Mat, then the following rules are used to determine the primary problem:

Rule 1: If $\mathrm{Tat}_{j}^{B} = Mat$ and sub-problem BSP^{j} includes multipurpose blend unit *j* then the primary problem is ($PP = \left\{BSP^{j}\right\}$).

Rule 2: If $\mathrm{Tat}_{j}^{B} = Mat$ and sub-problem BSP^{j} doesn't include multipurpose blend unit *j* then the blend scheduling sub-problem is chosen as the primary problem ($PP = \left\{BSP^{j}\right\}$).

Rule 3: If $\mathrm{Tat}_{j}^{B} \neq Mat$ and $\mathrm{Tat}^{P} = Mat$ then (PSP) is selected as the primary problem ($PP = \left\{PSP\right\}$).

When determining the primary problem, priority is given to a problem that satisfy rule 1; however if rule 1 is not satisfy then we check rule 2 and subsequently rule 3.

The solution of primary problem is used to determine parameter Ae_{n} which represents active event-point.

$$Ae_{n} = \begin{cases} 1 & if \sum_{i \in I_{j}} wv_{i,j,n} > 0 \\ 0 & if \sum_{i \in I_{j}} wv_{i,j,n} = 0 \end{cases}, \qquad \forall j \in J^{BC}, BSP^{j} \in PP \tag{36a}$$

$$Ae_{n} = \begin{cases} 1 & if \sum_{j' \in J^{PU}, j \in J_{j'}^{seq}, i \in I_{j'}, S_{j}^{c} \cap S_{i}^{p} \neq \varnothing} wv_{i,j',n} > 0 \\ 0 & if \sum_{j' \in J^{PU}, j \in J_{j'}^{seq}, i \in I_{j'}, S_{j}^{c} \cap S_{i}^{p} \neq \varnothing} wv_{i,j',n} = 0 \end{cases}, \qquad \forall j \in J^{BC}, PSP \in PP \tag{36b}$$

Parameter Ae_{n} is determined by equation (36a) if the primary problem is a sub-problem BSP^{j} otherwise it is determined by equation (36b).

Step 3: Using parameters $wv^a_{i,j}$, $in^a_{s,j,k}$, $out^a_{s,k,j}$, $\chi^a_{i,i',j}$, $\bar{\chi}^a_{i,i',j}$ and Ae_n we restrict the solution space of the relaxed sub-problems so that a solution of the restricted problem have production operations producing blend components at the same event-point as the blend units are mixing.. We fix binary variables { $wv_{i,j,n}$, $in_{s,j,k,n}$, $out_{s,k,j,n}$, $\chi_{i,i',j,n}$ } as shown equations (37a)-(37h).

$$wv_{i,j,n} = wv^a_{i,j} Ae_n, \quad \forall j \in J, j \notin J^m, i \in I_j, n \in N \tag{37a}$$

$$wv_{i,j,n} = 0, \quad \forall j \in J^m, i \in I_j, n \in N, \left(Ae_n = 0\right) \vee \left(wv^a_{i,j} = 0\right) \tag{37b}$$

$$\sum_{i \in I_j, wv^a_{i,j} > 0} wv_{i,j,n} = Ae_n, \quad \forall j \in J^m, n \in N, Ae_n > 0 \tag{37c}$$

$$\sum_{n, Ae_n > 0} wv_{i,j,n} \geq wv^a_{i,j}, \quad \forall j \in J^m, i \in I_j, wv^a_{i,j} > 0, \sum_{i',i''} \bar{\chi}^a_{i',i'',j} = 0 \tag{37d}$$

$$\chi_{i,i',j,n} = 0, \quad \forall j \in J, i \in I_j, i' \in I_j, n \in N, \left(Ae_n = 0\right) \vee \left(\chi^a_{i,i',j} = 0\right) \tag{37e}$$

$$\sum_{n, Ae_n > 0} \chi_{i,i',j,n} = \bar{\chi}^a_{i,i',j}, \quad \forall j \in J, i \in I_j, i' \in I_j, i \neq i', n \in N, \chi^a_{i,i',j} > 0 \tag{37f}$$

$$in_{s,j,k,n} = in^a_{s,j,k} Ae_n, \quad \forall k \in K, k \notin K^m, k \notin K_p, j \in J^{pk}_k, j \notin J^m, i \in I_j, s \in S_k, n \in N \tag{37g}$$

$$out_{s,k,k,n} = out^a_{s,k,j} Ae_n, \quad \forall k \in K, k \notin K^m, k \notin K_p, j \in J^{kp}_k, j \notin J^m, i \in I_j, s \in S_k, n \in N \tag{37h}$$

Equation (37a) fixes binary assignment variable for all units except for multipurpose units. For multipurpose units, equations (37b)-(37f) are used, where equation (37b) fixes assignment variable $wv_{i,j,n}$ to zero if the task i has not been active in relaxed problem or if the event-point n is not active in primary problem. Constraint (37c) forces unit to be active when $Ae_n = 1$ and constraint (37d) enforces that task i needs be active in unit j if $wv^a_{i,j} = 1$ and no tasks changeovers are happening $\sum_{i',i''} \bar{\chi}^a_{i',i'',j} = 0$ over the time horizon. Equation (37e) fixes continuous 0-1 variable $\chi_{i,i',j,n}$ to be zero at event-point n if $Ae_n = 0$ and the total number times the changeover from task i to i' can happen in unit j is fixed to $\bar{\chi}^a_{i,i',j}$ in constraint (37f). Binary variables $in_{s,j,k,n}$ and $out_{s,k,j,n}$ are fixed in restricted relaxed problem as shown in equations (37g)-(37h) for all units and tanks that do not perform multipurpose tasks and store final product tanks.

Since the blend products demand is to be satisfied before its due date, we do not fix any variables associated with the product tanks as not to restrict solution space of the restricted relaxed problem too much that product demands do not get satisfied. This situation can occur because in refinery under study, the products tanks can't load and unload products simultaneously. Thus, if we fix the binary variable associated with loading and unloading of product tanks based upon the solution of the relaxed problem, it may cause infeasibility in the restricted relaxed problem.

Restricted relaxed problem is built and solved for every sub-problem except for the sub-problem that is chosen as the primary problem. Solution of the restricted relaxed problem gives us consistent solution across (BSP) and (PSP) in terms of event-points so that the continuous process material balance is not violated.

5.4.2.2. Upper Bounding Problem

We fix many of the binary variables { $wv_{i,j,n}$, $in_{s,j,k,n}$, $out_{s,k,j,n}$, $l_{k,o,n}$ } in problem (IP) to the values of the corresponding binary variables obtained from the solution of restricted relaxed sub-problems and obtain a MILP model (IP-U). The solution of the restricted relaxed problem is { $wv^r_{i,j,n}$, $in^r_{s,j,k,n}$, $out^r_{s,k,j,n}$, $l^r_{k,o,n}$ } and based on non-zero values of these binaries, corresponding variables in (IP) are fixed. Only certain binary variables in problem (IP) are fixed to 1 and no binaries are fixed to 0 so that a better upper bound can be attained in earliest iterations in the LD algorithm. Equations (38a)-(38d) show how the variables are fixed in the problem (IP) to obtain a MILP model (IP-U).

$$w_{i,j,n} = 1, \quad \forall j \in J, i \in I_j, n \in N, wv^r_{i,j,n} > 0 \tag{38a}$$

$$in_{s,j,k,n} = 1, \quad \forall k \in K, j \in J^{pk}_k, i \in I_j, s \in S_k, n \in N, in^r_{s,j,k} > 0 \tag{38b}$$

$$out_{s,k,k,n} = 1, \quad \forall k \in K, j \in J^{kp}_k, i \in I_j, s \in S_k, n \in N, out^r_{s,k,j} > 0 \tag{38c}$$

$$l_{k,o,n} = 1, \quad \forall k \in K_p, j \in J^{pk}_k, i \in I_j, s \in S_k, n \in N, l^r_{k,o,n} > 0 \tag{38d}$$

Solving model (IP-U) yields an upper bound on the solution of (IP). The model (IP-U) is always feasible because the model includes demand, due-dates, and blend logistics giveaways.

5.5. Strengthening the Lagrangian Relaxation

There are several constraints, redundant or unnecessary in (IP) that can be added to the relaxed problem and the restricted relaxed problem to strengthen the performance of the Lagrangian algorithm. To improve the lower bound, we propose a pre-processing step before the Lagrangian algorithm and add constraints related to product demands to the decomposed problems in the algorithm. A common feature of the preprocessing steps and constraints strengthening lower bound is that they in some sense recreate what has been relaxed, namely the network structure of the refinery, but without destroying the separability of the Lagrangian sub-problems. Furthermore, using the solution of the best upper bound, in the decomposed problems, penalty variables relating violations of demands, due dates, and blend logistics can be fixed to zero.

5.5.1. Preprocessing Step

Because the refinery is decomposed into (BSP) and (PSP), production capacity limitation for the blend components in (PSP) is not taken into consideration while solving (BSP). Therefore, we present a preprocessing step that determines the maximum production rate for each blend component *s* and maximum production capacity for a set of blend components that can be mixed by a blend unit.

$$\text{minimize}\left\{ z_s^{PP} = -100 \sum_{j \in J^{PU}, j \in J_s^P, n \in N} Uof_{s,j,n} \right\}, \quad \forall s \in S_{bc} \tag{39}$$

$$\text{minimize}\left\{ z_j^{PP} = -100 \sum_{j' \in J^{PU}, s \in S_j^c \cap S_{j'}^P, n \in N} Uof_{s,j,n} \right\}, \quad \forall j \in J^{BU} \tag{40}$$

The preprocessing problem (PC-s) and (PC-j) are based upon the relaxed sub-problem corresponding to production unit operations (PSP). The model (PC-s) is obtained by replacing objective function of relaxed problem with that given in equation (71) to determine maximum production rate of each blend component. Another preprocessing problem (PC-j) is presented to calculate the maximum rate at which production units can supply materials to each blend unit.

The objective function for (PC-j) is presented in equation (72). Problem (PC-s) is solved for each blend component and problem (PC-j) for each blend unit before the Lagrangian algorithm begins. Furthermore these problems are independent of each other and can be solved in parallel.

Upon solving the pre-processing problems, we obtain values of variables $\left\{Cf_{s,j,t}^{P}, Ts_{j,t}^{P}, Tf_{j,t}^{P}, x_{j,n,t}^{P}\right\}$ $\forall s, j, n, t$ from the optimal solution.

$$RP_s^{\max} = \max_t \left\{ \frac{Cf_{s,j,t}^{P} \sum_n x_{j,n,t}^{P}}{Tf_{j,t}^{P} - Ts_{j,t}^{P}} \right\}, \quad \forall j \in J^{BU}, s \in S_j^c \tag{41a}$$

$$RP_j^{\max} = \max_t \left\{ \frac{\sum_{s \in S_j^c} Cf_{s,j,t}^{P} \sum_n x_{j,n,t}^{P}}{Tf_{j,t}^{P} - Ts_{j,t}^{P}} \right\}, \quad \forall j \in J^{BU} \tag{41b}$$

The maximum rate of production of component s is determined by equation (41a) using the optimal solution obtained upon solving corresponding preprocessing problem (PC-s). Similarly, the maximum rate for set of blend components supplied to blend unit j is determined by equation (41b) using the optimal solution values of corresponding preprocessing problem (PC-j).

5.5.2. Strengthening Lower Bound

To strengthen the Lagrangian decomposition algorithm, the information obtained in pre-processing step is applied to update the blend component production rate upper bounds in equations (13a), (13b), and (15) for relaxed and restricted relaxed blend scheduling problems. Computationally inefficient big M term is replaced with tighter bound upper $RP_s^{\max}$ as shown in equations (42a)-(42c).

$$Cf_{s,j,t}^{B} \le Uif_{s,j,n} + RP_s^{\max}\left(1 - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c, n \in N, t \in T, n \ge t \tag{42a}$$

$$Cf_{s,j,t}^{B} \ge Uif_{s,j,n} - RP_s^{\max}\left(1 - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c, n \in N, t \in T, n \ge t \tag{42b}$$

$$Cf_{s,j,t}^{B} \le RP_s^{\max} \sum_{n \ge t} x_{j,n,t}^{B}, \quad \forall j \in J^{BU}, s \in S_j^c, t \in T \tag{42c}$$

$$Cf_{s,j,t}^{B} \le RP_{s}^{\max}\left(Tf_{j,t}^{B} - Ts_{j,t}^{B}\right) + RP_{s}^{\max}UH\left(1 - \sum_{n \ge t} x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_{j}^{c}, t \in T \tag{42d}$$

$$\sum_{s \in S_{j}^{c}} Cf_{s,j,t}^{B} \le RP_{j}^{\max}\left(Tf_{j,t}^{B} - Ts_{j,t}^{B}\right) + RP_{j}^{\max}UH\left(1 - \sum_{n \ge t} x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, t \in T \tag{42e}$$

The amount of blend component processed by blend unit j at time-point t is bounded by the maximum production rate of the component and blend unit's process time as enforced by constraint (42d). Similarly, constraint (42e) states that the total bending rate is bounded by the maximum rate at which material is supplied to a blend unit j.

$$\sum_{i \in I_{s}^{p}, j \in J_{i}, n} wv_{i,j,n} \ge 1, \quad \forall s \in S_{A} \cup S_{B}, Amt^{\min} > 0 \tag{43}$$

$$\sum_{i \in I_{j}, i' \in I_{j}, i \ne i', n} \chi_{i,i',j,n} \ge -1 + \sum_{s \in S_{j}^{p}, i \in I_{s}^{p}, i \in I_{j}} 1, \quad \forall j \in J^{M} \cap J^{BU} \tag{44}$$

Here parameter $Amt^{\min} = \sum_{s \in S_{A}, o \in O_{s}} D_{o,s}^{-} - \sum_{s \in S_{A}, k \in K_{s}} sto_{s,k}$. Constraint (43) enforces that tasks that produce product s must be active at least once over the scheduling time horizon and constraint (44) enforces minimum number of changeover of tasks at multipurpose blend units. Constraint (43) is added to both (PSP) and (BSP); however, equation (44) is only present in (BSP).

$$\sum_{s \in S_{bc}, j \in J_{s}^{p}, n \ne N} Uof_{s,j,n} \ge Amt^{\min} - pen \tag{45}$$

$$\text{minimize}\left\{ L^{Pm}(u) = z + C^{11} pen - \sum_{j \in J^{BU}, t} u_{j,t}^{Ts} Ts_{j,t}^{P} - \sum_{j \in J^{BU}, t} u_{j,t}^{Tf} Tf_{j,t}^{P} - \sum_{j \in J^{BU}, s \in S_{j}^{c}, t} u_{s,j,t}^{Cf} Cf_{s,j,t}^{P} \right\} \tag{46}$$

Constraint (45) enforces that (PSP) must produce at least minimum amount of blend components required for (BSP) so that demand of blend products can be satisfied. If due to production capacity limitation, (PSP) cannot produce enough blend components, then we add positive penalty variable pen in equation (45) which is penalized in the objective function as shown in equation (46). The objective function of relaxed and restricted relaxed problems is now replaced by the one given in equation (46).

5.5.3. Construction of lower bound from the best upper bound

Due to production capacity limitation, the refinery might not be able to satisfy the demand of the products on time, might not respect all the blend logistics requirements, and may not be able to meet product quality specifications. To obtain a feasible schedule that can be implemented in the refinery, we have included, demand, due dates, and blend logistics giveaways in all scheduling models. These violations are heavily penalized in the objective function. Equation (47a) gives the demand and due violations cost and equation (47b) gives total violation cost by including the product downgrading and minimum heel violation penalty to the demand violation cost. We also obtain the cost of task changeovers for multipurpose blend units from feasible solution as given in equation (47c).

$$violatn_{dem} = \sum_{o \in O} c_o^{11} dg_o^l + \sum_{o \in O} c_o^{12} dg_o^u + \sum_{s \in S_B} c_s^{13} rg_s + \sum_{o \in O} c_o^{14} Tearly_o + \sum_{o \in O} c_o^{15} Tlate_o \tag{47a}$$

$$violatn_{total} = violatn_{dem} + \sum_{k \in K^m, s \in S_k, s' \in S_k, s \neq s', n} c_{k \in K^m}^{9} std_{s,s',k,n} + \sum_{k \in K^m, s \in S_k, n} c_{s,k}^{10} mh_{s,k,n} \tag{47b}$$

$$chgovr = \sum_{j \in J^m, i \in I_j, i' \in I_j, i' \neq i, n} c_{i,i'}^{6} \chi_{i,i',j,n} \tag{47c}$$

For a minimization problem, $Z^L \le Z^{opt} \le Z^U$ thus, if the best upper bound obtained at the current iteration provides a solution without any violations, then the optimal schedule of the original integrated problem will not have any violations. Furthermore, in refinery, an optimal solution with lowest task changeovers cost is desired if there are no demand and due-date violations. The desired changeovers at multipurpose blend units are from task producing high quality blend product to tasks producing low quality products so that the products produced during changeover transition are not discarded. Thus, we can conclude that $chgovr^{opt} \le chgovr^U$, if $violatn_{total}^U = 0$.

$$\left\{ dg_o^l, dg_o^u, rg_s, Tearly_o, Tlate_o \right\} = 0, \quad if \;\; violatn_{dem}^U = 0 \tag{48a}$$

$$\left\{ dg_o^l, dg_o^u, rg_s, Tearly_o, Tlate_o, std_{s,s',k,n}, mh_{s,k,n} \right\} = 0, \quad if \;\; violatn_{total}^U = 0 \tag{48b}$$

$$\sum_{j\in J^m, i\in I_j, i'\in I_j, i'\neq i, n} c^6_{i,i'}\chi_{i,i',j,n} \le chgovr^U, \quad if\ violatn^U_{total} = 0 \tag{48c}$$

We can produce tighter lower bound by considering value of parameter $violatn^U_{total}$ obtained from the best upper bound solution for a given set of Lagrangian multipliers. If best upper bound obtained satisfies finished products demand and their due-dates, $violatn^U_{dem} = 0$, then we fixed variables associated with demands and due-dates to zero as given by equation (48a). Similarly, in constraint (48b), variables associated with demand and blend logistics violations are fixed to zero if the best upper bound solution where $violatn^U_{total} = 0$. Furthermore, total changeover cost for multipurpose blend units is bounded by the total changeover cost ($chgovr^U$) using equation (48c).

5.6. Lagrangian Decomposition Algorithm

In this section, proposed Lagrangian decomposition algorithm is presented in detail. This algorithm differs from classical Lagrangian decomposition in the way that the lower bound and Lagrange multipliers are obtained. In classical Lagrangian decomposition (classical-LD), the lower bound is always obtained from the Lagrangian relaxed problem where as in the proposed restricted Lagrangian decomposition (restricted-LD) the tighter lower bound is chosen among the relaxed problem and the restricted relaxed problem.

Before presenting the details of the algorithm, the scheduling models used in the algorithm are defined first. The relaxed blend scheduling model (P1) consists of tasks, units, materials, tanks relating to (BSP) and is defined as below:

$$\text{minimize}\left\{ L^B(u) = z + \sum_{j\in J^{BU}, t} u^{Ts}_{j,t} Ts^B_{j,t} + \sum_{j\in J^{BU}, t} u^{Tf}_{j,t} Tf^B_{j,t} + \sum_{j\in J^{BU}, s\in S^c_j, t} u^{Cf}_{s,j,t} Cf^B_{s,j,t} \right\}$$

s.t. constraints (5-11),(12a-12d),(14a-14b),(16-17),(42a-42e),(43-44),(48a-48c),
and all the constraints corresponding to finished product blending
and delivery operations in model (IP) (P1)

The relaxed production unit scheduling model (P2) consists of tasks, units, materials, tanks relating to (PSP) and is defined as below:

$$\text{minimize}\left\{ L^{Pm}\left(u\right) = z + C^{11} pen - \sum_{j \in J^{BU}, t} u_{j,t}^{Ts} Ts_{j,t}^{P} - \sum_{j \in J^{BU}, t} u_{j,t}^{Tf} Tf_{j,t}^{P} - \sum_{j \in J^{BU}, s \in S_j^c, t} u_{s,j,t}^{Cf} Cf_{s,j,t}^{P} \right\}$$

s.t. constraints (18a-18b),(19-26),(27a-27b),(28a-28d),(31a-31c),(32a-32d),
(33-34),(43),(45),(48a-48c),
and all the constraints corresponding to production units scheduling operations in model (IP)

(P2)

The restricted relaxed models (P1-R) and (P2-R) are constructed by fixing binary variables using equations (37a)-(37h) in models (P1) and (P2), respectively. Infeasible restricted relaxed model is obtained by removing equations (48a)-(48c) from models (P1-R) and (P2-R). The upper bounding problem (IP-U) is obtained by fixing values of certain variables using equations (38a)-(38d) in full-scale model (IP) presented in chapter 4. The models (PC-s) and (PC-j) are obtained from (P2) by replacing its objective function by (39) and (40), respectively.

5.6.1. Updating the Multipliers

The subgradient method first proposed by Fisher (1981) is commonly used method to update Lagrangian multipliers used in solving Lagrangian relaxation problems. It requires solving all sub-problems at each iteration to find a subgradient of the relaxed problem to compose a search direction to update the multipliers. The elements of subgradient to the relaxed problems are as follows:

$$g_{j,t}^{Ts} = \overline{Ts}_{j,t}^{B} - \overline{Ts}_{j,t}^{P}, \quad \forall j \in J^{BU}, t \in T$$

$$g_{j,t}^{Tf} = \overline{Tf}_{j,t}^{B} - \overline{Tf}_{j,t}^{P}, \quad \forall j \in J^{BU}, t \in T$$

$$g_{s,j,t}^{Cf} = \overline{Cf}_{s,j,t}^{B} - \overline{Cf}_{s,j,t}^{P}, \quad \forall j \in J^{BU}, s \in S_j^c, t \in T$$

Where $\left(\overline{Cf}^{B}_{s,j,t}, \overline{Ts}^{B}_{j,t}, \overline{Tf}^{B}_{j,t}\right)$ and $\left(\overline{Cf}^{P}_{s,j,t}, \overline{Ts}^{P}_{j,t}, \overline{Tf}^{P}_{j,t}\right)$ are the values of the duplicating variables, obtained from the solution of the blend scheduling problem and production scheduling problem, respectively.

The multiplier is updated for the next iteration according to

$$u^{(m+1)} = u^{(m)} + s^{(m)} d^{(m)}$$

Here, $u^{(m)}$ denotes the multiplier values, $s^{(m)}$ is the step size, and $d^{(m)}$ is the search direction in iteration *m*. The search direction is usually set to the subgradient, but it is more efficient to use modified formula that takes into consideration pervious iteration direction. Search direction is defined as recursive formula as suggested in Gaivoronski (1988), where $d^{(1)} = g^{(1)}$:

$$d^{(m)} = \frac{\left(g^{(m)} + \theta d^{(m-1)}\right)}{\theta + 1}, \quad \forall m > 1$$

Where θ determines how much consideration is given to the previous direction, 0 being no consideration to previous direction and 1 being an average of current and previous search direction. The step size $s^{(m)}$ is given by a widely used formula:

$$s^{(m)} = \lambda^{(m)} \frac{\left(Z^U - L(u^{(m)})\right)}{\left\|g^{(m)}\right\|^2}$$

Here Z^U is the upper bound of (IP) , $L\left(u^{(m)}\right)$ is the solution of the relaxed problem at $u^{(m)}$, and $\lambda^{(m)}$ should be assigned a value in the interval ($\varepsilon, 2-\varepsilon$), where $\varepsilon > 0$ to ensure convergence. If there is no improvement in the lower bound in *K* successive iterations, we set $\lambda^{(m)} = 0.5\lambda^{(m-1)}$. $\lambda^{(m)}$ is reset back to $\lambda^{(0)}$ whenever an improved upper bound or lower bound.

5.6.2. Stopping Criteria

In this section, we present the stopping criteria used to terminate the Lagrangian algorithm.

- Stop if the feasible solution is found at lower bound, which happens when $g^{(m)} = 0$; but this rarely occurs for large problems.

- Stop if the lower bound from the Lagrangian relaxation exceeds the best known upper bound, i.e., if $Z^L \geq Z^U$.
- Stop if the duality gap is less than or equal to 1%.

$$duality\ gap = \left(\frac{Z^U - Z^L}{Z^L} \right)$$

- Stop if $\left\| d^{(m)} \right\| < \varepsilon$, $s^{(m)} < \varepsilon$, or $Z^U - Z^L \leq \varepsilon$.

In practice we do not wait till above mentioned stopping criteria is met, so we impose the number of iteration as the stopping criterion, i.e., stop after a limited number of iterations.

5.6.3. Algorithm

Here using $u^{(m)}$ at m^{th} iteration, the objective value of the relaxed problem and restricted relaxed problem is calculated from equation in (4b) and given as $Z_R\left(u^{(m)}\right)$ and $Z_{RR}\left(u^{(m)}\right)$ respectively. $Z^{(m)}$ is the objective value of the feasible solution through constructed heuristic at the m^{th} iteration.

The procedure for the restricted-LD algorithm:

Step 1: Solve pre-processing problems (PC-j) and (PC-s) as described in section 5.5.1

Step 2: Initialize m=0; $Z^U = +\infty; Z^L = -\infty; \left(u_{j,t}^{Ts}\right)^{(0)} = 0; \left(u_{j,t}^{Tf}\right)^{(0)} = 0; \left(u_{s,j,t}^{Cf}\right)^{(0)} = 0; \lambda^{(0)}; \theta$

Step 3: Solve relaxed problem by decomposition that is solve independent problems (P1) and (P2).

Obtain $Z_R, \left(\overline{Cf}_{s,j,t}^{B}\right)_R, violatn_{total}^{R}$, and $chgovr_R$.

Step 4: Construct a restricted relaxed problem as described in section 5.4.2. Solve the restricted sub-problems (P1-R) and (P2-R).

If infeasible, then solve corresponding infeasible sub-problem using model (P1-RI) or (P2-RI).

Obtain $Z_{RR}, \left(\overline{Cf}^{B}_{s,j,t}\right)_{RR}, violatn^{RR}_{total}$, and $chgovr_{RR}$.

Step 5: If $violatn^{RR}_{total} \le violatn^{R}_{total}$ and $Z_{RR}\left(u^{(m)}\right) > Z^{L}$, then $Z^{L} = Z_{RR}\left(u^{(m)}\right)$.

Otherwise, if $violatn^{RR}_{total} > violatn^{R}_{total}$ and $Z_{R}\left(u^{(m)}\right) > Z^{L}$, then $Z^{L} = Z_{R}\left(u^{(m)}\right)$.

Step 6: Construct a feasible solution to the original problem as described in section 5.4.2.2.

If $Z^{m} < Z^{U}$, then $Z^{U} = Z^{m}$ and obtain parameters $violatn^{U}_{dem}, violatn^{U}_{total}$, and $chgovr^{U}$ for constraints described in section 5.5.3.

Step 7: Update Lagrangian multipliers by using the sub-gradient method presented in section 5.6.1.

Step 8: Terminate the algorithm if the current solution satisfies at least one "stopping criteria" listed in section 5.6.2. is met. Otherwise, set $m = m+1$ and return to step 3.

In step 7, the Lagrangian multipliers are calculated using restricted relaxed problem solution unless $violatn^{RR}_{total} > violatn^{R}_{total}$. Problems in step 1 are independent and can be solved in parallel. Similarly, optimization models (P1) and (P2) in step 3 and models (P1-R) and (P2-R) in step 4 are independent of each other and can be solve in parallel to reduce computational time. The flowchart of the framework for the algorithm is given Figure 5.5.

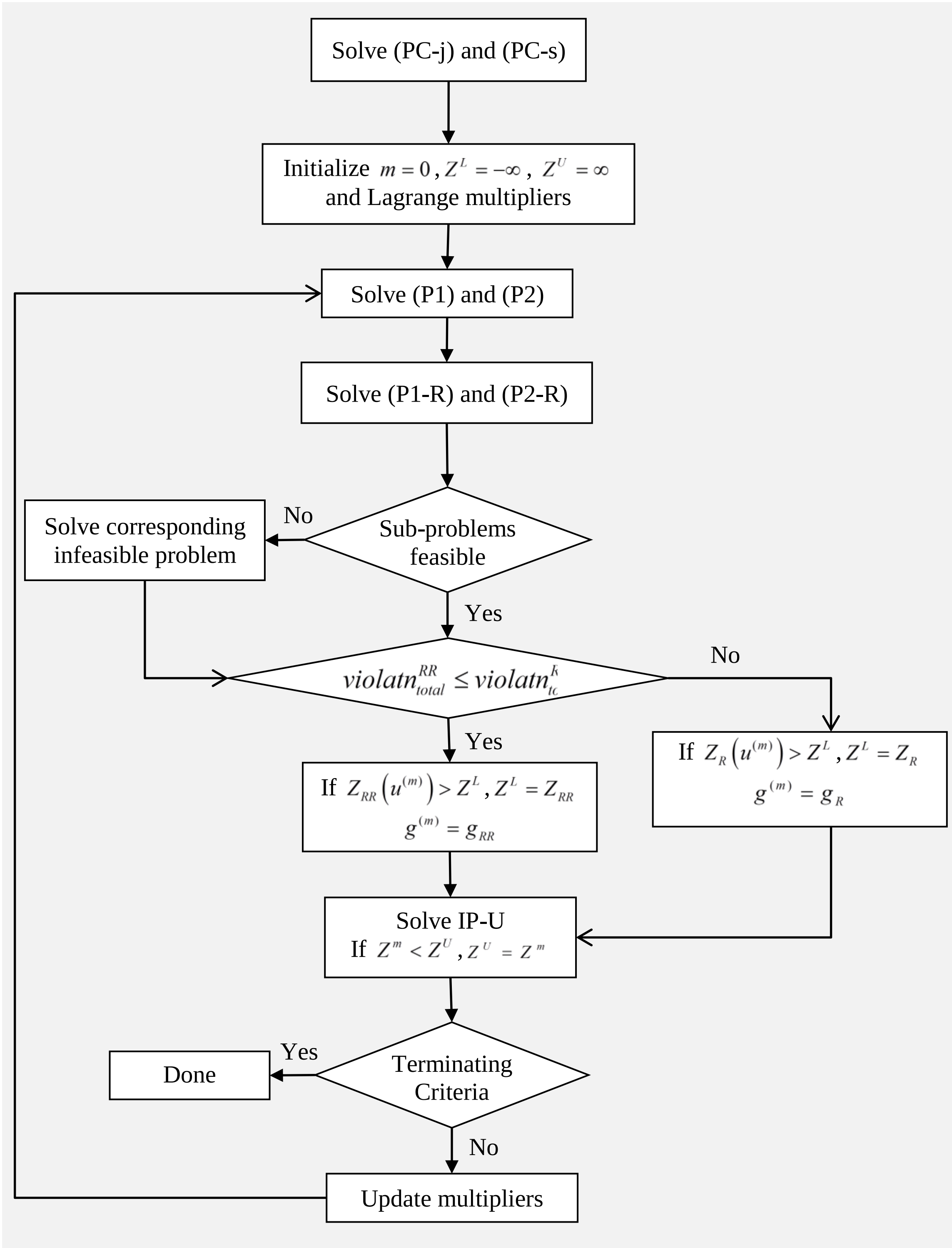

Figure 5.5. Algorithm for the proposed restricted-Lagrangian decomposition (restricted-LD)

5.7. Computational results

The computational results are obtained on Dell Precision (IntelR XeonTM with CPU 3.20 GHz, 3.19 GHz, and 2 GB memory) running on Linux using CPLEX 12.3.0/GAMS 23.7.2 to demonstrate their effectiveness in solving oil-refinery scheduling problems using 6 different examples, where each example differs in either demands values, intermediate due dates, and initial hold up in the tank. The Lagrangian algorithm is set up in MATLAB and interfaces with GAMS to solve scheduling problems. The algorithm is evaluated with three performance measures: duality gap, number of iterations, and computational time (seconds). The maximum solution time of 28,800 CPU seconds is used for full-scale problem and for the decomposed problem maximum solution time of 3,600 CPU seconds is used. Optimality tolerance of 1e-6 is used as termination criteria for all examples solved using CPLEX. For the algorithm, the maximum number of iteration is set to 100, $\varepsilon = 1^{-6}$, and K = 2.

Table 5.1 Model statistics for preprocessing problems

	(PC-s)	**(PC-j)**	
		Diesel Blender	**Jet Blender**
Event pt.	3	3	3
Int./Cont. var.	120/924	120/980	120/978
(Constraints)	(2425)	(2425)	(2425)
Nonzero Elem.	7753	7813	7813

The preprocessing problems for each blend components and blend units are solved before Lagrangian algorithm is implemented. The model statistics for preprocessing models are presented in Table 5.1 and these models are solved in less than 1 CPU seconds. The model statistics for full-scale integrated model and relaxed sub-problems for 6 different examples are shown in Table 5.2. Total number variables in restricted relaxed sub-problems, (P1-R) and (P2-

R), are same as that of corresponding relaxed models because (P1-R) and (P2-R) are obtained by restricting solution space by including additional constraints and fixing values of certain binaries/continuous 0-1 variables in corresponding relaxed models, (P1) and (P2).

Table 5.2 Model statistics

Ex.	**# Orders**	**Full Scale Model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.	**Blend scheduling model (P1)** Event pt. Int./Cont. variables (Constraints) Nonzero Elem.		**Production scheduling model (P2)** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.
			Diesel Blender	**Jet Blender**	
1	4	5 360/2663 (6387) 22633	5 127/958 (2489) 9819	5 33/247 (717) 2323	5 200/1526 (4649) 15394
2	4	5 360/2663 (6379) 22569	5 127/958 (2481) 9760	5 33/247 (717) 2329	5 200/1526 (4649) 15405
3	4	5 360/2663 (6379) 22569	5 127/958 (2480) 9725	5 33/247 (717) 2324	5 200/1526 (4649) 15395
4	6	5 380/2771	5 139/1022	5 41/291	5 200/1526

		(6741) 24018	(2695) 10598	(847) 2794	(4648) 15399
5	8	6 484/3453 (8644) 31765	6 184/1309 (3649) 14956	6 60/402 (1272) 4384	6 240/1835 (5794) 19442
6	13	7 736/5005 (13394) 53145	7 277/1829 (5585) 23804	7 83/530 (1833) 6643	7 280/2146 (7027) 23903

The computational results for full-scale model and Lagrangian algorithm are shown in Table 5.3. Proposed restricted-Lagrangian decomposition algorithm is quite effective for these refinery scheduling problems and we obtain good solutions to the scheduling problem. Figure 5.6 to Figure 5.11 shows the convergence of upper and lower bound using the classical LD and restricted LD algorithms in examples 1-6, respectively. The proposed restricted-LD algorithm outperforms classical Lagrangian decomposition in terms of quality of the solution, duality gap, number of iterations, and computational time. In classical approach, the lower bound is obtained solely from relaxed problem's solution, whereas proposed restricted-LD picks the best lower bound between the solution of relaxed problem and restricted relaxed problem. Thus, restricted-LD lower bounds take into consideration the continuous process characteristic of the refinery units and gives better lower bound as observed in Figure 5.6 to Figure 5.11.

In most cases, both algorithms provide an upper bound closer to the optimal solution of original problem in first iteration because feasible problem is constructed using restricted relaxed problem solutions and has flexibility of obtaining a better solution because only the certain binaries are fixed to 1 and none are fixed to 0. In situation where first iteration doesn't provide an upper bound closer to optimal solution, the upper bound improves vastly early on and ultimately provides a better solution. In example 5, the first iteration provides an upper bound that is very far from an optimal solution, however, it improves greatly in second iteration, and by fifth iterations it's almost closed to the optimal value. After finding the best upper bound early on, both algorithms spend rest of the computational time proving optimality.

Table 5.3 Computational Results

Ex	**Full Scale Model**			**Classical LD** $\lambda = 1.50, \theta = 0.8$				**Restricted LD** $\lambda = 1.20, \theta = 0.7$			
	z	Gap (%)	CPU sec	z	Gap (%)	CPU sec	Total Iter	z	Gap (%)	CPU sec	Total Iter
1	450.77	0.00	342	450.77	29.34	652	43	450.77	-0.21	113	11
2	598.64	0.00	367	598.64	5.92	2133	57	622.91	0.15	251	9
3	6222.54	0.00	8952	6222.54	1.49	15304	71	6222.54	0.79	3245	14
4	592.81	0.00	1535	602.33	25.22	1048	26	592.81	0.95	1225	40
5	503.19	0.00	6458	504.25	12.62	1234	22	503.86	7.72	2652	20
6	749.35	72.8	28800	552.50	18.04	36865	20	512.65	-0.86	20490	15

The performance (duality gap) of restricted-LD algorithm is better than that of classical approach because restricted-LD is able to improve lower bound faster than classical LD. Proposed restricted-LD provides better duality gap than classical LD and restricted-LD is terminated in almost all cases when gap is less than 1% as shown in Table 5.3. In case of classical decomposition approach, the algorithm was terminated for all examples when the step size meets the predefined tolerance of 1e^{-6}. The step size reaches the tolerance faster because we cut the step size parameter $\lambda^{(m)}$ by half if the lower bound does not improve after 2 iterations. Even when the duality gap is high, the upper bound is closer to the original problem optimal solution in both algorithms.

Table 5.4 Time spent (%) in each step compared to the total solution time of Lagrangian decomposition algorithms

	Problems	**Ex. 1**	**Ex. 2**	**Ex. 3**	**Ex. 4**	**Ex. 5**	**Ex. 6**
Classical LD	Relaxed	77.32	91.47	98.28	90.78	94.16	93.31
	Restricted relaxed	11.48	4.40	0.96	4.17	2.80	2.95
	Feasible	11.20	4.13	0.75	5.05	3.04	3.74
Restricted LD	Relaxed	68.12	87.92	98.76	89.61	97.48	91.96
	Restricted relaxed	13.26	4.94	0.49	4.19	1.31	7.96
	Feasible	18.63	7.14	0.75	6.19	1.21	0.076

Table 5.4 compares the computational effort exerted to solve relaxed problem, restricted relaxed problem, and feasible upper bounding problem. As expected, majority of the time is spent in solving relaxed problem. Classical LD takes 652 CPU seconds to solve example 1 and spends 77.32% of the total time on solving for relaxed problem and only 11.48% on solving restricted relaxed problem. Similarly for restricted-LD, example 1 takes 68.12% of the total computational time on solving relaxed problem, 13.26% on solving restricted relaxed problem, and 18.63% on solving upper bounding problem MILP problem. Even for the large scale complex problem given in example 6, the upper bounding MILP problem can be easily solved. Computational time can be further improved by solving relaxed sub-problems in parallel and similarly restricted relaxed problems can be solved in parallel.

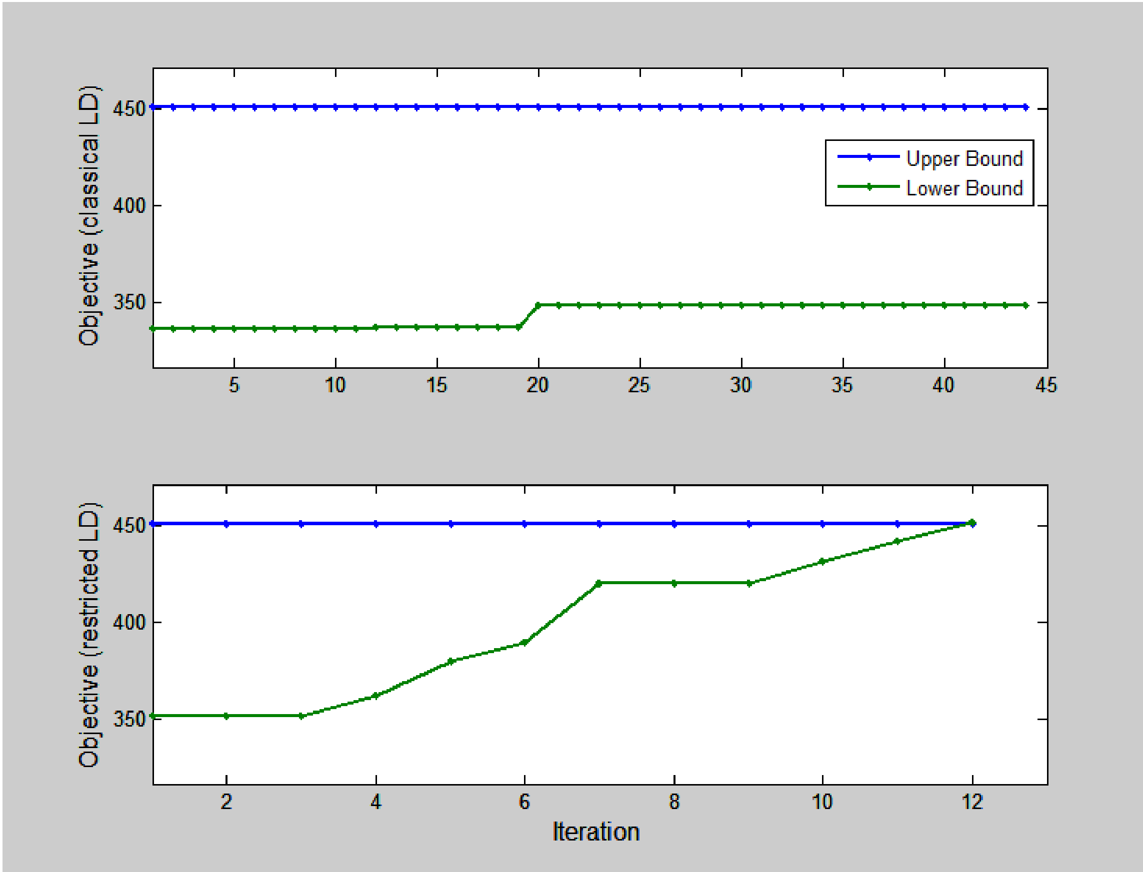

Figure 5.6 Convergence of upper and lower bound of Lagrangian decomposition for example 1.

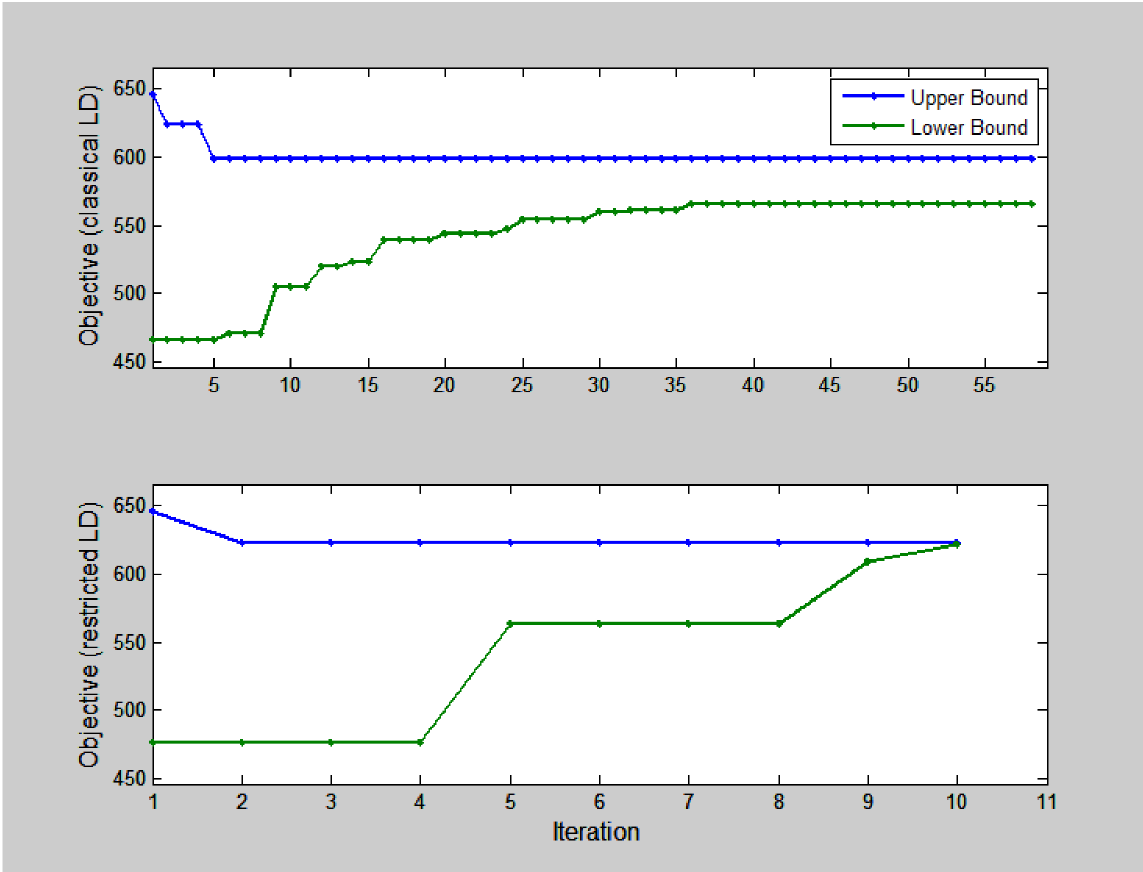

Figure 5.7 Convergence of upper and lower bound of Lagrangian decomposition for example 2.

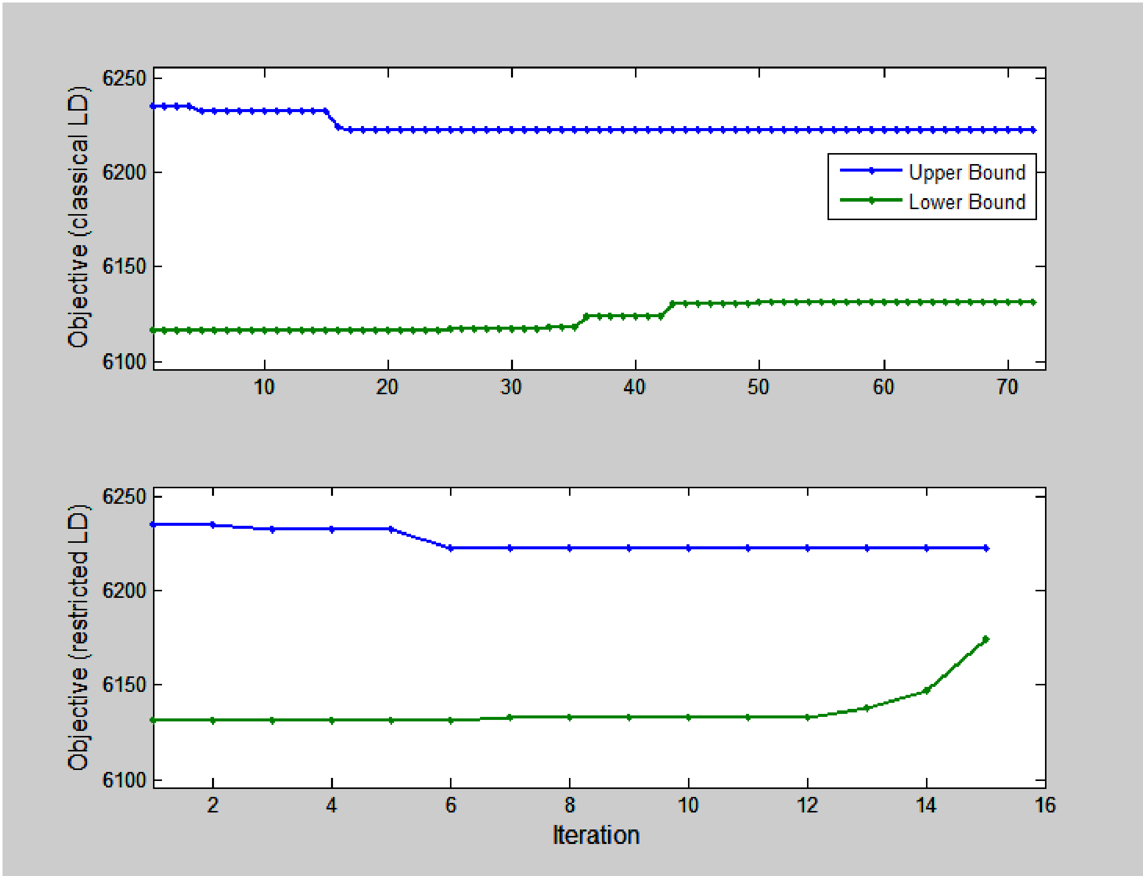

Figure 5.8 Convergence of upper and lower bound of Lagrangian decomposition for example 3.

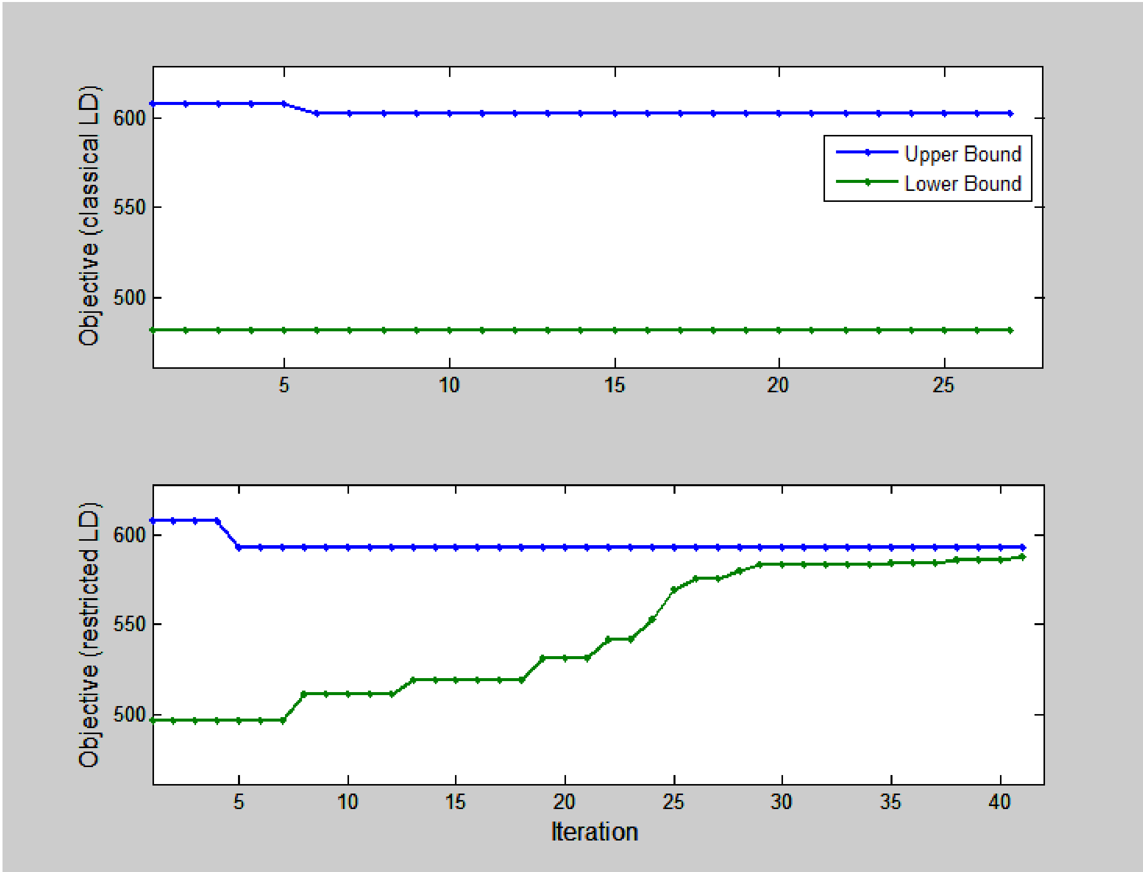

Figure 5.9 Convergence of upper and lower bound of Lagrangian decomposition for example 4.

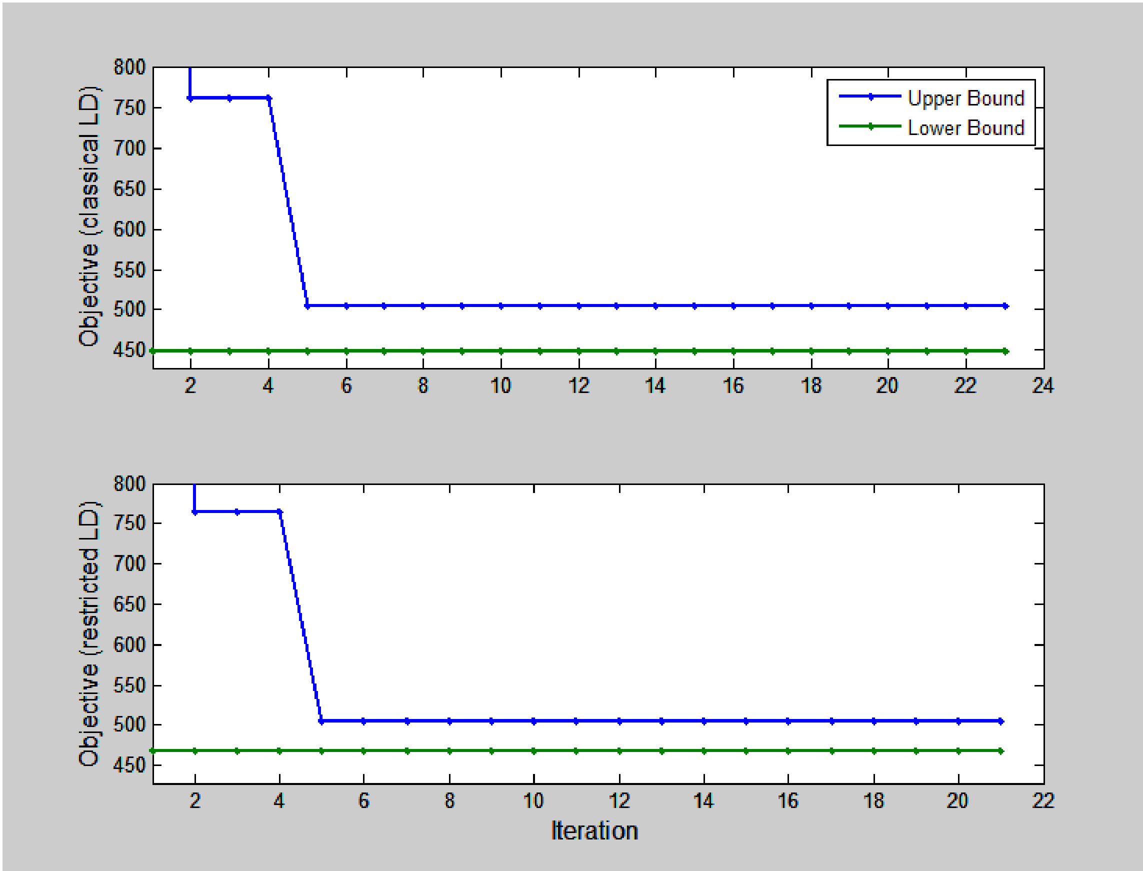

Figure 5.10 Convergence of upper and lower bound of Lagrangian decomposition for example 5.

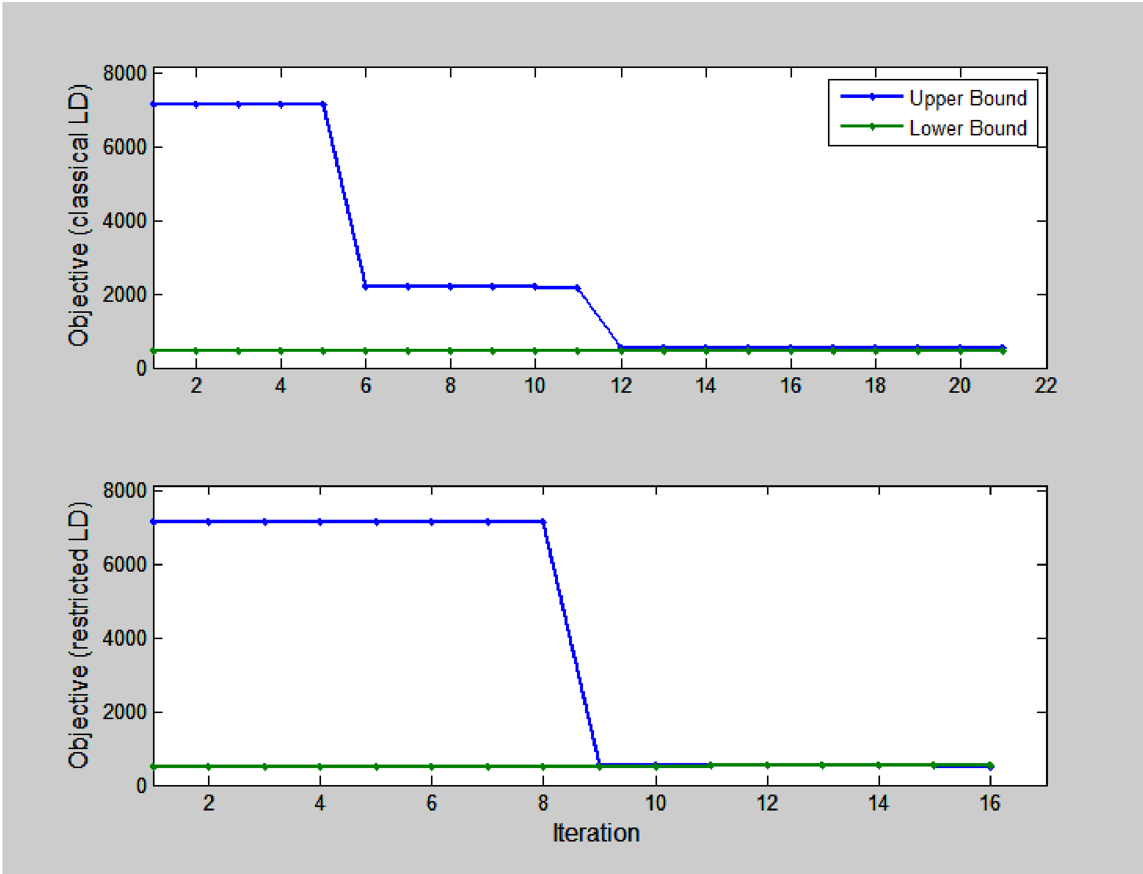

Figure 5.11 Convergence of upper and lower bound of Lagrangian decomposition for example 6.

5.8. Summary

This chapter introduces a Lagrangian decomposition framework, restricted Lagrangian relaxation problems for integrated production unit scheduling and finished product blending and delivery scheduling problems. The algorithm is built based on the mathematical formulation given in Chapter 4. A restricted relaxed Lagrangian problem is constructed to facilitate generation of tighter upper bound in each iteration. Restricted relaxed sub-problems are based upon the solution of Lagrangian relaxed sub-problems and solution of restricted relaxed problem takes into consideration the continuous process characteristic of rundown streams. To improve the performance of the algorithm, a preprocessing step, constraints for decomposed sub-problems, and inclusion of the best upper bound's solution in lower bounding problems are proposed. The proposed restricted relaxed sub-problems produce better lower bounds and better upper bounds. Computational results of a real case study show that the proposed algorithm is very effective and provide better solutions in reasonable times.

Nomenclature

Indices

i	Tasks
j	Production units
k	Storage tanks
n	Event-points
m	Lagrangian algorithm iteration counter
o	Product order
p	Properties
s	States
t	Continuous time-points similar to event-points

Sets

I_j	Tasks which can be performed in unit *j*
I_s^c / I_s^p	Tasks which can consume/produce material s
J	All Production units
J^{BU}	Blend units
J^{PU}	Production units that belong to production unit operations
J_s^c / J_s^p	Units that consume/produce material *s*
J_i	Units which are suitable for performing task *i*
J^h	Units that can produce all the same products as some other unit in the

	refinery
J^{m}	Units which are suitable for performing multiple tasks
J_j^{seq}	Units that follow unit j (no storage in between)
J_k^{kp} / J_k^{pk}	Units that consume/produce material s stored in tank k
J^{P}	Units that produce products
K	Storage tanks
K^{h}	tanks that can store the same products as some other tank in the refinery
K_j^{kp} / K_j^{pk}	Tanks that store material consumed/produce by unit j
K^{m}	Multipurpose tanks that can store multiple materials
K_p	Tanks that can store final products
K_s	Tanks that can store material s
N	Event-point within the time horizon
O	Orders for products that are stored in tanks
O_s	Orders for finished product s
P	Product Properties
S	States
S_A	Group A products, stored in tanks
S_B	Group B products, not stored in tanks
S_{bc}	States that are blended by blend-units
S_k	States that can be stored in tank k
S_i^c / S_i^p	States that can be consumed/produced by task i
S_j^c / S_j^p	States that can be consumed/produced by unit j

Parameters

$D^{+}_{o,s}, D^{-}_{o,s}$	Demand limit requirement for order *o* and product *s* that is stored in tank
$\max\_rate_s$	Maximum rate of production of material *s*
$ot_{o,i}$	1 if task *i* is performed when order *o* of PUSP is processed by BSP.
r_s	Demand of the final product s at the end of the time horizon
$RB_j^{\min} / RB_j^{\max}$	minimum and maximum production rate of a blend unit *j*
$RP_j^{\max}$	Maximum rate material is supplied by (PSP) to blend unit *j*
$RP_s^{\max}$	Maximum rate blend component *s* is supplied by (PSP) to blend unit *j*
$R_{i,j}^{\min} / R_{i,j}^{\max}$	Minimum/ maximum rate of material be processed by task *i* in unit *j*
RL_i	Minimum run length for task *i*
$RU_k^{\min} / RU_k^{\max}$	Minimum/maximum rate of product unloading at tank *k*
$sto_{s,k}$	Amount of state *s* that is present at the beginning of the time horizon in *k*
ta_k	Fill draw delay for product tank *k*
UH	Available time horizon
$V_k^{\max}$	Maximum available storage capacity of storage tank *k*
V_k^{heel}	Maximum heel available for storage tank *k*
$yo_{s,k}$	1 if the material *s* is present at the beginning of the time horizon in k
$\rho_{s,i}^{\min} / \rho_{s,i}^{\max}$	Proportion of state *s* produced/consumed by task *i*
$\rho b_{s,j}^{\min} / \rho b_{s,j}^{\max}$	Minimum/maximum proportion of blend component *s* consumed by blend unit *j*

Variables

Binary Variables

$wv_{i,j,n}$	Assignment of task *i* in unit *j* at event-point *n*
$in_{s,j,k,n}$	Assigns the material flow of *s* into storage tank *k* from unit *j* at point *n*

$l_{k,o,n}$	Assigns the starting of product flow out of product tank k to satisfy order o at event-point n
$out_{s,k,j,n}$	Assigns the material flow of s out of storage tank k into unit j at point n
$y_{s,k,n}$	Denotes that material s is stored in tank k at event-point n

0-1 continuous variables

$\alpha_{j,n}$	For unit j, 1 if the unit becomes active for very first time at event-point n
$\beta_{k,n}$	For tank k, 1 if the tank becomes active for very first time at event-point n
$\eta_{s,s',k,n}$	1 if material in tank k switchover service from s at event-point n to s' at later event-point
$\eta o_{s,s',k}$	1 if material in tank k switchover service from s to s'
$\chi_{i,i',j,n}$	1 if task at unit j changes from i at event-point n to i'at later event-point.
$a_{j,n}$	Assigns the materials flow to blend unit j at event-point n
$x_{j,n,t}$	Denotes that blend unit j is active at event-point n and time-point t

Positive variables

$bp_{s,i,j,n}$	Amount of material s produced task i in unit j at event-point n
$bc_{s,i,j,n}$	Amount of material s undertaking task i in unit j at event-point n
$Cf^{B}_{s,j,t}$ / $Cf^{P}_{s,j,t}$	Amount of blend component s consumed/supplied at blend unit j at time-point t
dg^{l}_{o}	Minimum demand quantity give-away term for order o
dg^{u}_{o}	Maximum demand quantity give-away term for order o
H	Total time horizon used for production tasks
$JJf_{s,j,j',n}$	Flow of state s from unit j to consecutive unit j' for consumption at point n
$Kif_{s,j,k,n}$	Flow of material s from unit j to storage tank k event-point n

$Kof_{s,k,j,n}$	Flow of material s from storage tank *k* to unit *j* at point *n*
$Lf_{o,k,n}$	Flow of final product for order *o* from storage tank *k* at event-point *n*
$mh_{s,k,n}$	Maximum heel give-away term for product tanks
rg_s	Minimum demand quantity give-away term for Group B product *s*
$Rif_{s,k,n}$	Flow of raw material to storage tank *k* event-point *n*
$st_{s,k,n}$	Amount of state *s* present in storage tank *k* at event-point *n*
$std_{s,s',k,n}$	Amount of state *s* that is downgraded to state s' in storage tank *k* at event-point *n*
$Tearly_o$	Early fulfillment of order *o* than required
$Tf_{i,j,n}$	Time that task *i* finishes in unit *j* at event-point *n*
$Tf^B_{j,t} / Tf^P_{j,t}$	Finish time for materials consumption/supply at blend unit *j* at time-point *t*
$Tlate_o$	Late fulfillment of order *o* than required
$Tos_{k,o,n}$	Time that material starts to flow from tank *k* for order o at event-point *n*
$Tof_{k,o,n}$	Time that material finishes to flow from tank *k* for order o at event-point *n*
$Ts_{i,j,n}$	Time that task *i* starts in unit *j* at event-point *n*
$Ts^B_{j,t} / Ts^P_{j,t}$	Start time for materials consumption/supply at blend unit *j* at time-point *t*
$Tsf_{j,k,n}$	Time that material finishes to flow from unit *j* to tank *k* at event-point *n*
$Tsf_{k,j,n}$	Time that material finishes to flow from tank *k* to unit *j* at event-point *n*
$Tss_{j,k,n}$	Time that material starts to flow from unit *j* to storage tank *k*
$Tss_{k,j,n}$	Time that material starts to flow from tank *k* to unit *j* at event-point *n*
$Uif_{s,j,n}$	Flow of raw material *s* to production unit *j* at point *n*
$Uof_{s,j,n}$	Flow of product material *s* from unit *j* at point *n*
$UUf_{j,s,v,n}$	Amount of state *s* received by unit *j* at period *v* is processed at event *n*

z	Objective value of full-scale integrated model

Chapter 6

6. Efficient Decomposition Approach for Large Scale Refinery Scheduling

This chapter proposes a heuristic algorithm for integrated scheduling of refinery production unit operations and finished product blending and delivery operations. This work addresses the interdependences of the production unit operations and finished product blending and delivery operations in the refinery that has components streams coming directly from process units to blend headers with no components storage options available. To apply the proposed iterative algorithm, the integrated oil-refinery scheduling problem is decomposed into production unit scheduling problem and into finished product blending and delivery scheduling problem. The goal of the algorithm is to obtain a feasible solution that satisfies products demands commitments with minimum demurrage violations while minimizing inventory cost and penalties subject to product and logistics giveaways. The applicability of the algorithm is illustrated by a realistic large-scale refinery case study producing diesel and jet fuel. Significant savings are realized in the computational effort.

6.1. Introduction

Even with inclusion of valid inequalities the refinery operations scheduling model is still computationally prohibitive. The mathematical decomposition approach presented in previous chapter can lead to global optimal solution, however for practical application where the need for feasible solution is greater than obtaining the best solution possible, the solution times of mathematical decomposition is undesirable. For this situation, heuristics approach exploiting inherent structure of the problem alongside insight into the makeup of a feasible solution provides an alternative to massive full-scale optimization problem or over mathematical decomposition. Furthermore, heuristics approach can be used as preprocessing step for mathematical decomposition solution strategy to reduce complexity of the original model (i.e. reduce feasible solution space). (Amaldi et al., 2008; Arief et al., 2007; Cornillier et al., 2008; Karuppiah et al., 2010; Kelly, 2003b; Roslof et al., 2002)

Usually blend component tanks are present when the blending and delivery sub-problem is addressed without simultaneously addressing production unit scheduling problem. However, when blend components tanks are not available, blend units are directly connected to the upstream production units. In order to comply with the finished blend product properties specifications, blend recipe needs be determined by considering the production constraints of upstream processes so that the blend components production and consumption rate is consistent across two scheduling decision levels. These interdependences of the blending operations and production unit scheduling operations require an integrated approach to the refinery scheduling.

In this work, we decomposed the large-scale integrated problem into a production unit scheduling problem and a finished product blending and delivery problem. A heuristic algorithm is proposed based on the decomposed network to obtain a good solution of the original integrated problem in reasonable computational times. The algorithm is built upon the mathematical formulation developed given in Chapter 4. The proposed decomposition approach focuses on effectively solving the decomposed problem to obtain a feasible solution that satisfies demands while minimizing due date violations, product giveaway, task changeovers at multipurpose blend units and tanks, and inventory costs.

The outline of the chapter is as follows. Section 6.2 presents additional valid inequalities, section 6.3 presents proposed solution approach and the heuristic algorithm is discussed in section 6.4. A realistic case study presented in section 4.2.1 is used to illustrate the applicability of the proposed algorithm to a large-scale model in section 6.6.

6.2. Mathematical Formulations

Mathematical formulations proposed earlier for the oil-refinery operations scheduling problem is based on the continuous time representation and an idea of unit-specific event points. (M. G.

Ierapetritou & C. A. Floudas, 1998; M.G. Ierapetritou & C.A. Floudas, 1998; Ierapetritou et al., 1999) State-task network (STN) representation introduced by Kondili et al. (1993) is used in formulating the problem. Event based formulation enables the representation of time in a continuous manner without unnecessary time slots or intervals. Although the location of the event points is unknown, the number of event points has to be considered initially. It is not straightforward to know the minimum number of event points needed to achieve the global optimum solution. Choosing a large number of event points makes it more likely to achieve the global optimum solution but it also results in larger solution times. A number of advances of the original methodology have been proposed in the literature to improve the scheduling formulation. (Janak et al., 2004; Mouret et al., 2011) The model involves material balance constraints, capacity constraints, demand constraints, quality constraints, logistics constraints, and setup constraints. The demand constraints ensure that all the products demand are satisfied while the quality constraints ensures that product quality specifications are met. To preserve the linearity of the formulation, linear blending rules are considered. The logistics constraints include all the operational rules presented in the previous section and the setup constraints models the activation of parallel production units and parallel storage tanks during the scheduling horizon. Due to production capacity limitation, downgrading of a higher quality product to a lower quality product may be necessary to satisfy demand on time and constraints are added in the model to formulate product downgrading in multipurpose storage tanks. If a feasible solution that satisfies all the quantity and logistics constraints cannot be obtained, then it is essential to produce a schedule that can still be implemented in real-life oil-refinery sacrificing model feasibility. In this case, we introduced artificial variables to treat any infeasibility present and these variables are subsequently penalized in the objective function to obtain an optimal solution that satisfies as many as possible from the quantity logistics constraints by minimizing giveaways.

The objective function (1) is used to maximize the performance, minimize inventory costs and product demands and intermediate due-dates violations.

$$
\begin{aligned}
z = & \sum_{i,j\in J_i,n} c^1_{i,j} wv_{i,j,n} + \sum_{k\in K_p,o,n} c^2_k l_{k,o,n} + \sum_{s,k\in K_s} c^3_k st_{s,k,N} + \sum_{j\in J^h,n} c^4_j \alpha_{j,n} + \sum_{k\in K^h,n} c^5_k \beta_{k,n} \\
& + \sum_{j\in J^m, i\in I_j, i'\in I_j, i'\neq i, n} c^6_{i,i'} \chi_{i,i',j,n} + \sum_{k\in K^m, s\in S_k, s'\in S_k, s\neq s'} c^7_{s,s'} \eta o_{s,s',k} + \sum_{k\in K^m, s\in S_k, s'\in S_k, s\neq s', n} c^8_{s,s'} \eta_{s,s',k,n} \\
& + \sum_{k\in K^m, s\in S_k, s'\in S_k, s\neq s', n} c^9_k std_{s,s',k,n} + \sum_{k\in K^m, s\in S_k, n} c^{10}_{s,k} mh_{s,k,n} + \sum_{o\in O} c^{11}_o dg^l_o + \sum_{o\in O} c^{12}_o dg^u_o \\
& + \sum_{s\in S_B} c^{13}_s rg_s + \sum_{o\in O} c^{14}_o Tearly_o + \sum_{o\in O} c^{15}_o Tlate_o
\end{aligned}
\tag{1}
$$

The oil-refinery performance is represented by longer run-modes ($wv_{i,j,n}$, $l_{k,o,n}$) and minimization of inventory ($st_{s,k,n}$), start up set-ups ($\alpha_{j,n}, \beta_{k,n}$), changeovers ($\chi_{i,i',j,n}, \eta o_{s,s',k}, \eta_{s,s',k,n}$), maximum heel violations ($mh_{s,k,n}$) and production downgrading ($std_{s,s',k,n}$). Logistics giveaways, under and over production, and demurrage are penalized. The penalty weights are assigned arbitrarily to each term depending on its importance in schedule. The deviation from intermediate due dates ($Tearly_o, Tlate_o$) and under/over production of finished products (dg^l_o, dg^u_o, rg_s) are heavily penalized. It is favorable that the multipurpose product tanks store different grade products only in a certain order that is allowed by the sequence dependent switchovers constraint. The favorable switchovers are from higher grade of product to lower grade and unfavorable switchovers are from lower grade to higher grade product. Due to contamination issues, unfavorable switchovers are more heavily penalized than favorable. Similar to multipurpose tanks, there is a sequence dependent switchover restriction for multipurpose blend units. Note that the different penalty parameters have significant effect on the computational time required to obtain an optimal solution.

All these constraints and objective function give rise to a large scale, incomprehensible mixed integer linear programming (MILP) model that becomes computationally expensive to solve using standard optimization software to optimality.

Valid Inequalities

To improve the computational efficiency, valid inequalities for the refinery scheduling model are introduced in previous chapter. These valid inequalities are based on certain logistics and operational rules and material balance. These inequalities are redundant to the original model because the optimizer will eventually discards infeasible solution and arrive at a feasible solution. Thus, these inequalities do not cut feasible solutions space but rather eliminate the region of infeasible solution explicitly instead of waiting for optimizer to reach that conclusion. Application of valid inequalities in scheduling model to speed up convergence to optimal solution can be found in work of Velez et al. (2013), G. K. D. Saharidis and Ierapetritou (2009), and Khor et al. (2012).

The valid inequalities that are included in the full-scale integrated model used in this work are as follows:

Constraint (1a) connects total amount of material consumed and produced at unit j and event point n in a form of inequality since material balance constraints present in the fullscale model uphold material balance requirements.

$$\begin{aligned}&\sum_{k\in K_j^{kp}\cap K_s} Kof_{s,k,j,n} + \sum_{j'\in J_j^{seq}\cap J_s^{p}} JJf_{s,j',j,n} + Uif_{s,j,n} \le \\ &\qquad \sum_{k\in K_j^{pk}\cap K_s} Kif_{s,j,k,n} + \sum_{j'\in J_j^{seq}\cap J_s^{c}} JJf_{s,j,j',n} + Uof_{s,j,n}, \quad \forall j\in J, n\in N\end{aligned} \tag{1a}$$

Constraints (1b-1d) reaffirms binary variable $y_{s,k,n}$ to be 1 if there is material s flowing in and out of the storage tanks k at given event point n.

$$\sum_{j\in J_k^{pk}} in_{s,j,k,n} \le \sum_{j\in J_k^{pk}} y_{s,k,n}, \quad \forall k\in K, s\in S_k, n\in N \tag{1b}$$

$$\sum_{j\in J_k^{kp}} out_{s,j,k,n} \le \sum_{j\in J_k^{kp}} y_{s,k,n}, \quad \forall k\in K, s\in S_k, n\in N \tag{1c}$$

$$\sum_{o \in O_s} l_{k,o,n} \le y_{s,k,n}, \quad \forall k \in K_{bp}, s \in S_k, n \in N \tag{1d}$$

If the tank is sending or receiving material *s* from a production unit *j* at event point *n*, then that unit will be active at event point *n* and these allocation requirements are represented by equations (1e-1f).

$$\sum_{k \in K_j^{pk} \cap K_s} in_{s,j,k,n} \le \sum_{k \in K_j^{pk} \cap K_s} \sum_{i \in I_j \cap I_s^p} wv_{i,j,n}, \quad \forall j \in J^{pk}, s \in S_j^p, n \in N \tag{1e}$$

$$\sum_{k \in K_j^{kp} \cap K_s} out_{s,k,j,n} \le \sum_{k \in K_j^{kp} \cap K_s} \sum_{i \in I_j \cap I_s^c} wv_{i,j,n}, \quad \forall j \in J^{kp}, s \in S_j^c, n \in N \tag{1f}$$

Constraints (1g-1h) asserts material *s* flow in/out of tanks to be active at event point *n* if the unit *j* is processing the material at that event point.

$$\sum_{k \in K_j^{pk} \cap K_s} in_{s,j,k,n} \ge \sum_{i \in I_j \cap I_s^p} wv_{i,j,n}, \quad \forall j \in J^{pk}, s \in S_j^p, n \in N \tag{1g}$$

$$\sum_{k \in K_j^{kp} \cap K_s} out_{s,k,j,n} \ge \sum_{i \in I_j \cap I_s^c} wv_{i,j,n}, \quad \forall j \in J^{kp}, s \in S_j^c, n \in N \tag{1h}$$

In addition to the valid inequalities proposed in our earlier work, we propose here some new valid inequalities to further improve the computational performance.

If two units are consecutive without any storage tank between them, then constraint (1i) affirms the simultaneous operation of these units due to the continuous operation mode. However, this constraint is not imposed on parallel production units that can produce the same type of products. For units that follow or are followed by parallel units, valid inequalities (1j-1l) are included.

$$\sum_{i \in I_j} wv_{i,j,n} = \sum_{i \in I_{j'}} wv_{i,j',n}, \quad \forall j \in J / J^h, j' \in J / J^h, j \ne j', j' \in J_j^{seq}, n \in N \tag{1i}$$

$$\sum_{i \in I_j} wv_{i,j,n} \le \sum_{j' \in J^h, j \in J_{j'}^{seq}, i \in I_{j'}} wv_{i,j',n}, \quad \forall j \in J^h, n \in N, \text{ and } \exists j' \in J^h, j \in J_{j'}^{seq} \tag{1j}$$

$$\sum_{i \in I_j} wv_{i,j,n} \le \sum_{j' \in J^h, j \in J_{j'}^{seq}, i \in I_{j'}} wv_{i,j',n}, \quad \forall j \in J / J^h, n \in N, \text{ and } \exists j' \in J^h, j \in J_{j'}^{seq} \tag{1k}$$

$$\sum_{i \in I_j} wv_{i,j,n} \le \sum_{j' \in J^h \cap J_j^{seq}, i \in I_{j'}} wv_{i,j',n}, \quad \forall j \in J / J^h, n \in N, J_j^{seq} \cap J^h \ne \varnothing \tag{1l}$$

As mentioned earlier, the demand order set O is arranged according to the ascending due date start time. That is, $times_{o'} \ge times_o, \quad \forall o, o' \ne o, o' > o$. Operational requirement for the finished blend product tank is that loading and unloading cannot happen at the same time, thus if the material is unloaded at tank k at event point $n+1$ then the tank should not be empty at previous event point n. This is condition is captured by equation (1m).

$$\sum_{o} l_{k,o,n+1} \le \sum_{s \in S_k} y_{s,k,n}, \quad \forall k \in K_{bp}, n \in N, n < N \tag{1m}$$

The demand order set O is arranged in chronologically ascending order based on intermediate due-dates. We apply inequality (1n) to require that the demand order o to be satisfied by product tank k earlier than order $o' > o$. Furthermore, if the initial inventory of the products is less than the required total minimum demand orders, then the production should take place before the demand is fulfilled. This requirement is upheld by constraint (1o) where $pr_s = \sum_{o \in O_s} D_{o,s}^{-} - \sum_{k \in K_s} sto_{s,k}$.

$$l_{k,o',n} \le \sum_{k' \in K_s, n' \le n} l_{k',o,n'}, \quad \forall k \in K_{bp}, s \in S_k, o \in O_s, o' \in O_s, o' > o, n \in N \tag{1n}$$

$$l_{k,o,n} \le \sum_{i \in I_s^p, j \in J_i, n' < n} wv_{i,j,n'}, \quad \forall s \in S_A \cap S_{bp}, k \in K_s, o \in O_s, pr_s > 0, n \in N, n \ne 1 \tag{1o}$$

The mathematical formulation presented in presented in last chapter, inequalities (1a-1o), and objective function (1) comprise the large-scale MILP model for an integrated scheduling problem.

6.3. Solution Strategy

In this section, we propose a heuristics-based algorithm for solving full-scale integrated problem within reasonable computational times. The algorithm involves solving independent sub-problems in an iterative fashion until a feasible solution is found. Spatial decomposition is applied to refinery system under study as shown in Figure 6.1, to obtain smaller sub-problems that are easier to solve than full-scale integrated problem. (Jia & Ierapetritou, 2003; Karuppiah et al., 2008) The refinery system under study involves rundown blending operations, thus overall refinery structure is decoupled by splitting the blend components stream originating from process units operations and supplying directly to blend headers. This decoupling produces two scheduling sub-problems relating to the production unit operations and blending and delivery operations of finished products, respectively. The complicating variables associated with split pipelines are defined differently in decomposed scheduling sub-problems than are in the integrated problem. Production unit scheduling problem defines variables associated with component streams as demand variables whereas blend scheduling operations defines the component flow variables as raw materials variables, as seen in Figure 6.1. Hence, the main goal of the production unit scheduling sub problem (PSP) is to satisfy the demand requirement of final products that belong to set S_B and demand of the blend components required by the blend scheduling sub problem. Similarly, the purpose of the blend scheduling sub problem (BSP) is to satisfy the demand of the finished blend products by mixing the raw materials (blend components), supplied by PSP, following the blend recipe and product property specifications.

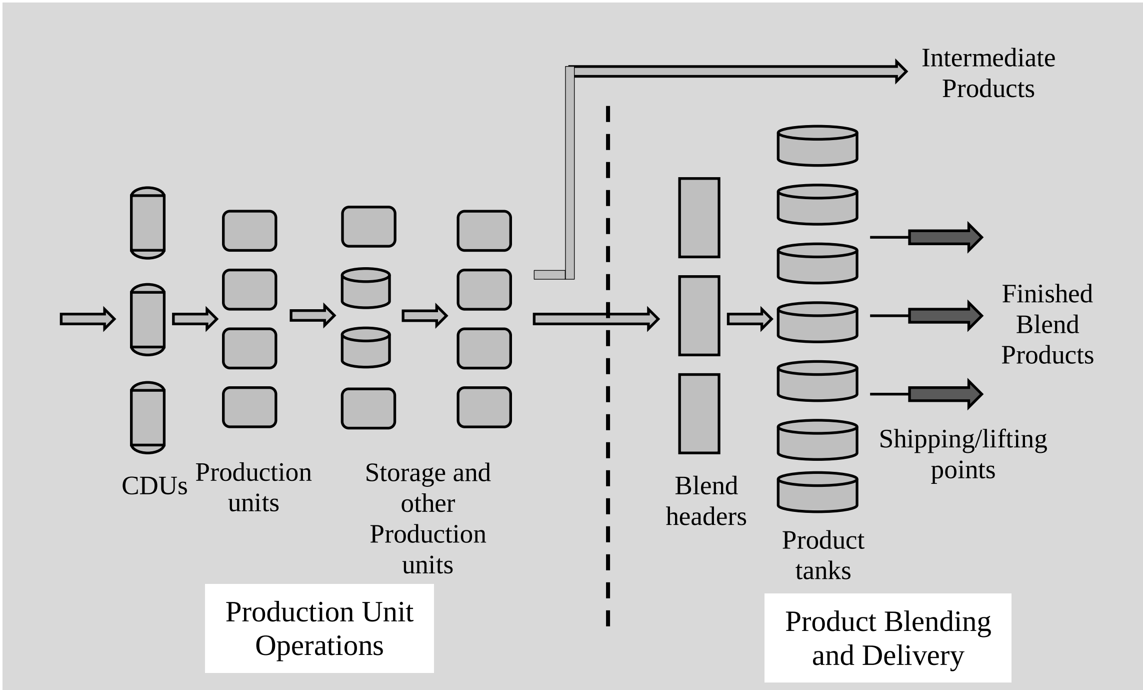

Figure 6.1 Decomposition of an oil-refinery operations network by splitting blend components pipelines

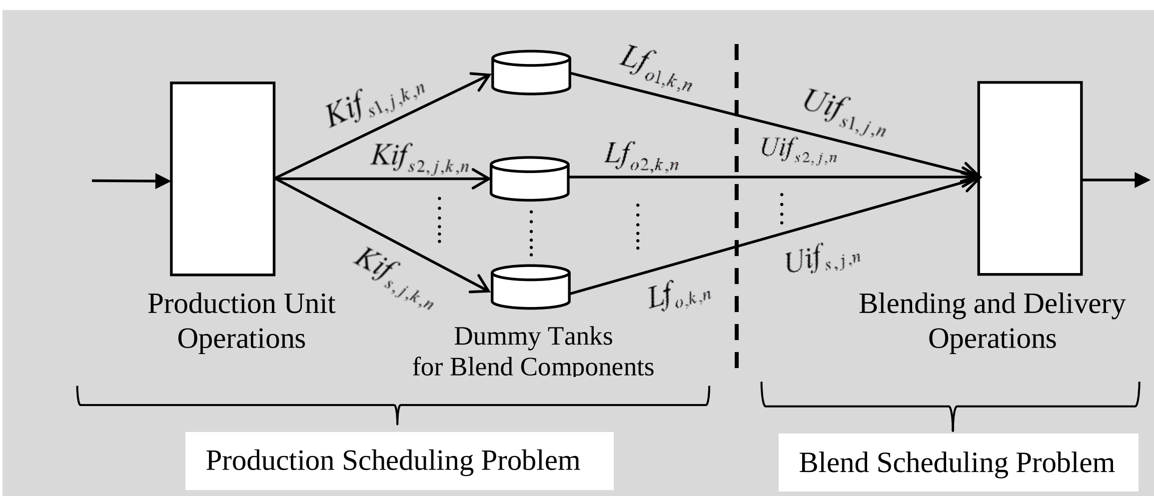

Figure 6.2 Illustration of scheduling sub-problems obtained by splitting component flow streams connecting process units directly to blend headers

In refinery system under study, blend components flow from process units to blend headers happen simultaneously with blending operations. Thus, in decomposed system, production unit scheduling problem needs to satisfy demand of blend components within certain time frame. We introduce dummy tanks for blend components in PSP as seen in Figure 6.2 for ease of imposing strict intermediate due-dates for components demand. These dummy tanks acts as product storage tanks in PSP but have different operational rules than finished blend product tanks. These operational rules as follow:

- Each dummy tank are dedicated to a blend component
- Simultaneous loading and unloading of blend components
- No accumulation of component at the end of an event-point
- Dummy tanks can supply components to multiple blend headers at same time
- Multiple tanks supplying directly to the same blend header must do so at the same time

Constraints relating to these rules are presented in detail later, but first, a general outline of the proposed iterative algorithm is presented in Figure 6.3. The algorithm solves BSP, PSP and restricted BSP in iterative fashioned until a global feasible solution that satisfies the final products demand requirements while minimizing demurrage. Here, the restricted blend scheduling problem (RBSP) is derived by limiting the feasible solution space of BSP based on solutions of PSP and BSP.

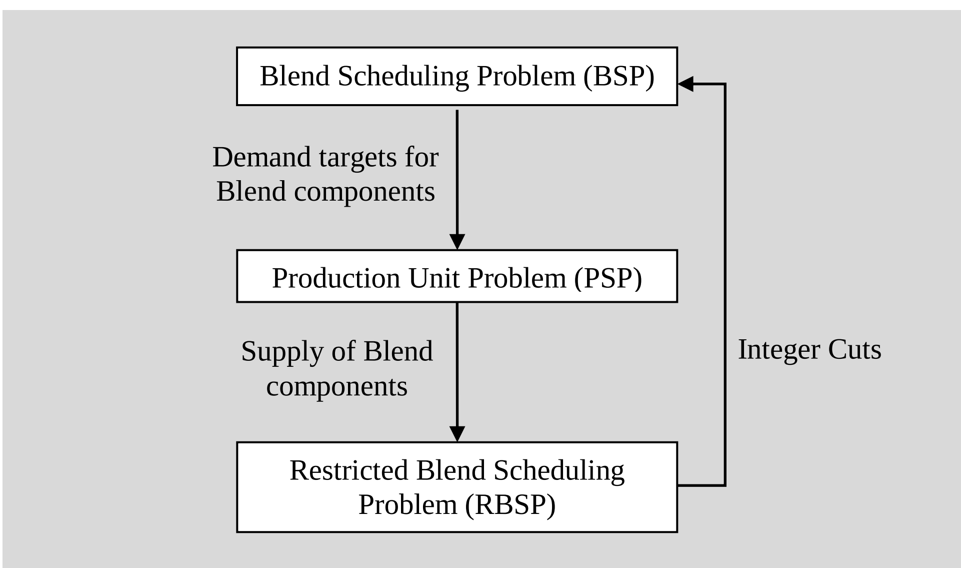

Figure 6.3 Proposed solution approach.

The details of BSP, PSP, and RBSP are presented in the following sub-sections. These problems include all the constraints present in the full-scale model and some additional constraints that are necessary to facilitate the convergence to a global feasible solution at the end of iteration. The demand constraint is given in equation (2) and a feasible global solution of the heuristic algorithm must satisfy the minimum demand requirements hence demand give-away variables $dg_o^l = 0$ and $rg_s = 0$ in equation (2).

$$D_{o,s}^{-} + r_s - dg_o^l - rg_s + \le \sum_{k,n} Lf_{k,o,n} + \sum_{j,n} Uof_{s,j,n} \le D_{o,s}^{+} + dg_o^u, \quad \forall s \in S_b, o \in O_s \cup \forall s \in S_f \tag{2}$$

6.3.1. Blend Scheduling Problem

Blend scheduling problem (BSP) involves blend headers, finished blend products storage tanks, delivery of these products according to intermediate due-dates and raw materials being supplied directly to blend headers. Refinery may include more than one blend headers producing multiple blend products. In the event that refinery has more than one blend header, we can further decompose BSP into smaller scheduling sub-problems depending upon the interactions between blend headers. The blend scheduling sub-problems must not share common blend components and finished product tanks and should produce different types of finished blend products. Hence, blenders using shared components must be jointly optimized. Decomposed independent blend scheduling sub-problems can be solved in parallel.

The mathematical model for BSP contains all the constraints that are present in the original model presented in chapter 4 and some additional constraints. To obtain a solution that provides minimum amount of blend component necessary for satisfying finished product demands, we fix the maximum finished blend-product demand to the minimum demand requirement, that is, $D^{+}_{o,s} = D^{-}_{o,s}$. Furthermore, the demand give-away variables are fixed to zero, $dg^{l}_{o} = 0, dg^{u}_{o} = 0$. Thus, the demand constraint is rewritten as follow for BSP:

$$\sum_{k \in K_s, n} Lf_{k,o,n} = D^{-}_{o,s}, \quad \forall s \in S_A, o \in O_s \tag{3a}$$

Where, parameter $D^{-}_{o,s}$ .is a minimum demand of product *s* and order *o* and variable $Lf_{k,o,n}$ is the flow of product from product tank *k* to satisfy to order *o* at event-point *n*. Furthermore, instead of having amount of product satisfied by tank *k* to be bounded by minimum and maximum unloading rate, $RU^{\min}_{k}$ and $RU^{\max}_{k}$, as is in full-scale model; for BSP, unloading rate is fixed to the maximum and the capacity constraints given in original model can be restated as follow:

$$Lf_{k,o,n} = RU^{\max}_{k}\left(Tof_{k,o,n} - Tos_{k,o,n}\right), \quad \forall k \in K_p, o \in O, n \in N \tag{3b}$$

To consider the production capacity of the process units producing blend components, maximum blend components flow rate constraint (3c) is included. The maximum blend components flow rate (max_rate_s) parameter is determined a priori using the maximum production recipe and maximum production capacity of the process units.

$$Uif_{s,j,n} \le \text{max\_rate}_{\text{s}}\left(Tf_{i,j,n} - Ts_{i,j,n}\right) + UH\left(1 - wv_{i,j,n}\right), \quad \forall j \in J^{b}, i \in I_j, s \in S^{c}_{i}, n \in N \tag{3c}$$

Furthermore, we can reduce the feasible solution space by not allowing the production to take place at the very last event point since the blend scheduling problem must respect the loading and unloading restrictions of product tanks, which means that material produced at the last event would not contribute to satisfying the demand and would just accumulate.

$$wv_{i,j,n} = 0, \quad \forall j \in J^{b}, i \in I_j, n \in N, n = N \tag{3d}$$

Restriction given in equation (3d) reduces the computational cost involve solving BSP without sacrificing an optimal solution of BSP.

6.3.1.1. Acquiring Parameters for PSP and RBSP from Solution of BSP

The optimal solution of the BSP provides the information regarding the amount ($Uif_{s,j,n}$) of blend component *s* flowing into each blend header *j* at event point *n* to produce blend product *s* using task *i*. The start and finish time ($Ts_{i,j,n}$ *and* $Tf_{i,j,n}$) of the blend component flow into the unit is also available. Based on this information, we derive demand, due dates, and other parameters for the production unit scheduling problem.

Each blend header gives rise to $\left|S_{bc}^{j}\right| \cdot n^{j}$ demand orders $o \in O^{j}$ that needs to be satisfied by PSP. Here, $S_{bc}^{j} = S_{j}^{c} \cap S_{bc}$ and $\left|S_{bc}^{j}\right|$ is the cardinality of the set S_{bc}^{j}, $n^{j} = \sum_{i,n} wv_{i,j,n}$, and O^{j} is a set of demand orders comprising of blend components consumed by blend unit *j*. Parameter $MN = \max_{j}\left(n^{j}\right) + 1$. The parameter $ot_{o,i}$ is 1 if task *i* is performed by a blend header *j* when order *o* is processed. A parameter $at_{o} = 1$ for the demand orders determined using the solution of BSP. The active event points of each blend unit *j* are used to assign values to demand orders as: $D_{o,s}^{-} = Uf_{s,j,n}$, $D_{o,s}^{+} = Uf_{s,j,n}$, $times_{o} = Ts_{i,j,n}$, $timef_{o} = Tf_{i,j,n}$, and $ot_{o,i} = wv_{i,j,n}$. That is, if a given blend header is active for 3 event points and can blend 3 components, a total of 9 active demand orders correspond to the blend header, as seen in Figure 6.4.

In order to take into consideration the interactions of the production units in PSP and to allow flexibility for over production, $\left|S_{bc}^{j}\right|$ additional demand orders are included to the set O^{j}. These additional demand orders are also necessary to satisfy group B product demands and to meet blend component due-dates without causing infeasibility in PSP. These demand orders have values of $D_{o,s}^{-} = 0$, $D_{o,s}^{+} = M$, $times_{o} = 0$, $timef_{o} = UH$. Parameter $ot_{o,i}$ for these additional demand orders is equal to the task mode of the last active event point (n^j) so that the task changeovers cost is minimized. Each demand order only corresponds to one blend component, thus the cardinality of the order set O for PSP is $\sum_{j \in J^B} \left|S_{bc}^j\right| \cdot \left(n^j + 1\right)$. Demand orders greater than $\left|S_{bc}^j\right| \cdot n^j$ do not need to be satisfied by PSP and are called non-active demand orders and $at_o = 0$. In Figure 6.4, demand orders $o10$, $o11$, and $o12$ are non-active demand orders.

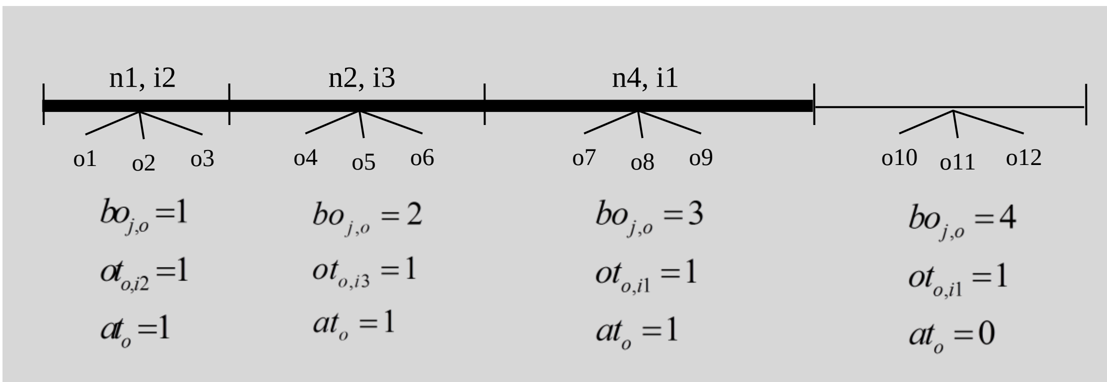

Figure 6.4 Determining parameters for PSP using a solution of a blend header *j*

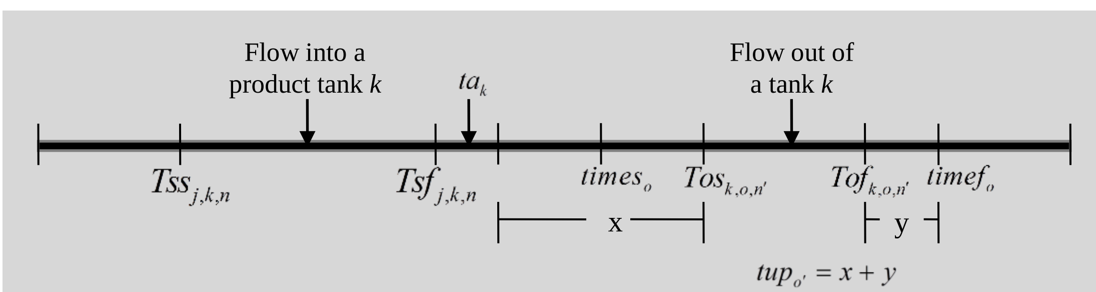

Figure 6.5 Determining parameter tup_o for PSP using the corresponding blend header *j*

The in-line blender mixes multiple components at the same time to produce final products, thus we introduce a parameter $bo_{j,o}$ to capture the relation between different demand orders for a blend unit *j*. As mentioned before, each blend unit has $\left|S_{bc}^j\right|\left(n^j + 1\right)$ demand orders belonging to set O^j. We assign the value of 1 to the first set of demand orders that are processed at the same time by the blend unit *j*. For the last set of demand orders, $bo_{j,o} = n^j + 1$. Note that the set O^j is arranged in an ascending chronological order based on due-date. Figure 6.4 shows how active demand orders are assigned and how parameter $bo_{j,o}$ is determined.

To minimize demurrage violations for finished blend-products in the optimal solution of RBSP, a feasible solution of PSP should satisfy blend components demand within certain time frame. For this purpose, a parameter tup_o is determined from the optimal solution of BSP. A product tank *k* receives finished product produced by blend unit *j* to satisfy finished product demand within the due date window. From the optimal solution of BSP, we know which tank receives the product produced by the blend unit when processing PSP blend component demand order *o* and we can determine the very first finished product demand order *o* that is satisfied by the product tank. We use the end time ($Tsf_{j,k,n}$) of the material flow into the tank *k*, start time ($Tos_{k,o,n}$), finish time ($Tof_{k,o,n}$) and a due date ($times_o$ - $timef_o$) of the first demand order *o* satisfied by the tank k to determine $tup_{o'}$ as shown in Figure 6.5. Here, parameter ta_k is fill-draw-delay for the product tank and $tup_{o'}$ is the maximum demurrage violation that can be incurred by blend component demand order o' that is satisfied by PSP without demurrage violation of finished product due date in RBSP. Another parameter that is determined from the optimal solution of BSP is $\bar{\chi}_{j,i,i'}$. Parameter $\bar{\chi}_{j,i,i'}$ is 1 if the changeover of service from task *i* to *i'* at blend unit *j* happens at any time during the scheduling horizon, $\sum_n \chi_{i',i,j,n} > 0$, otherwise it is 0. The parameters $D^-_{o,s}$, $D^+_{o,s}$, $times_o$, $timef_o$, $ot_{o,i}$, $bo_{j,o}$, at_o, tup_o, and $\bar{\chi}_{j,i,i'}$ are obtained from the optimal solution of BSP and are used in formulating the production unit scheduling problem and at least $(MN-1)$ event points are needed to solve the PSP in order to obtain a feasible solution.

6.3.2. Production Unit Scheduling Problem

Goal of production unit scheduling (PSP) is to satisfy demands of intermediate products (S_B) and blend components. To this end, PSP is solved using the parameters determined from the optimal solution of BSP. Formulation for PSP follows the original model expect that few new constraints are added and some existing constraints are eliminated to obtain a feasible solution that satisfy group B products demand ($rg_s = 0$). The oil-refinery system under study in this work does not possess blend components storage tanks. However, to model the intermediate due dates for blend components, we have introduced $|S_{bc}|$ dedicated storage tanks for blend component products. These artificial product tanks have zero accumulation at the end of each event-point as given by equation (4a). Thus, material balance constraint for tanks present in Chapter 4 is rewritten as equation (4a') for blend components tanks.

$$st_{s,k,n} = 0, \quad \forall s \in S_{bc}, k \in K_s, n \in N \tag{4a}$$

$$st_{s,k,n} = \sum_{j \in J_k^{pk}} Kif_{s,j,k,n} - \sum_{o \in O_s} Lf_{k,o,n}, \quad \forall s \in S_{bc}, k \in K_s, n \in N \tag{4a'}$$

In the full-scale model, the finished blend product tanks have the restriction of not loading material out of a tank when there is a material flowing into the tank. For PSP, to satisfy material balance of the blend component tanks, it is imperative to have simultaneous flow of material into and out of these tanks. Furthermore, since material flow in and out of these tanks should happen at the same time, we introduce (4b-4e) sequence constraints to align the start and finish time of flow in and out of the blend component tank.

$$Tss_{j,k,n} \geq Tos_{k,o,n} - UH\left(2 - l_{k,o,n} - in_{s,j,k,n}\right), \quad \forall s \in S_{bc}, k \in K_s, o \in O_s, j \in J_k^{pk}, n \in N \tag{4b}$$

$$Tss_{j,k,n} \leq Tos_{k,o,n} + UH\left(2 - l_{k,o,n} - in_{s,j,k,n}\right), \quad \forall s \in S_{bc}, k \in K_s, o \in O_s, j \in J_k^{pk}, n \in N \tag{4c}$$

$$Tsf_{j,k,n} \geq Tof_{k,o,n} - UH\left(2 - l_{k,o,n} - in_{s,j,k,n}\right), \quad \forall s \in S_{bc}, k \in K_s, o \in O_s, j \in J_k^{pk}, n \in N \tag{4d}$$

$$Tsf_{j,k,n} \leq Tof_{k,o,n} + UH\left(2 - l_{k,o,n} - in_{s,j,k,n}\right), \quad \forall s \in S_{bc}, k \in K_s, o \in O_s, j \in J_k^{pk}, n \in N \tag{4e}$$

During rundown blending operations, blend unit must receive all the blend components at the same time and this requirement is captured by equations (4f-4h). Constraints (4g-4h) are included because demurrage violation variables $Tlate_o$ and $Tearly_o$ are not be penalized in the objective function but a positive variable $late_o$ defined in the equation (4i) is penalized.

$$\sum_{k \in K_p} l_{k,o,n} = \sum_{k \in K_p} l_{k,o',n}, \quad \forall j \in J^b, o \in O^J, o' \in O^J, o \neq o', bo_{j,o} = bo_{j,o'}, n \in N \tag{4f}$$

$$Tlate_o = Tlate_{o'}, \quad \forall j \in J^b, o \in O^J, o' \in O^J, o \neq o', bo_{j,o} = bo_{j,o'} \tag{4g}$$

$$Tearly_o = Tearly_{o'}, \quad \forall j \in J^b, o \in O^J, o' \in O^J, o \neq o', bo_{j,o} = bo_{j,o'} \tag{4h}$$

$$Tlate_o \le tup_o + late_o, \quad \forall o \in O, at_o = 1 \tag{4i}$$

If multiple blend headers share common blend component, then at any given time, blend component can supply material to multiple blend units. However, a product tank cannot supply multiple demand orders to the same blend unit at the same time. This restriction is captured by constraint (4j).

$$\sum_{o \in O_s, bo_{j,o} > 0} l_{k,o,n} \le 1, \quad \forall k \in K_p, s \in S_{bc} \cap S_k, j \in J_s^c \cap J^b, n \in N \tag{4j}$$

Demand violations ($dg_o^l \ge 0, dg_o^u \ge 0$) for blend components are not subject to penalty in the PSP objective function. However constraint (4k) ensures that the total demand for a given blend unit is satisfied for rundown blending operation at time period *v*.

$$\sum_{s \in S_{bc}, o \in O_s \cap O^j, k \in K_s, at_o = 1, bo_{j,o} = v} Lf_{k,o,n} \ge \sum_{s \in S_{bc}, o \in O_s \cap O^j, at_o = 1, bo_{j,o} = v} D_{o,s}^-, \quad \forall j \in J^b, v \in \{1, 2, ..., MN\} \tag{4k}$$

Constraint (4l) takes maximum production capacity of a blend unit into consideration by limiting total flow of blend components to the blender.

$$\sum_{o'\in O,k\in K_p,bo_{j,o'}=v} Lf_{k,o',n} \le R_{i,j}^{\max}\left(Tof_{k,o,n} - Tos_{k,o,n}\right) + UH\left(1 - l_{k,o,n}\right),$$
$$\forall j \in J^b, s \in S_j^c, o \in O_s, k \in K_s, i \in I_j, ot_{o,i} = 1, v \in \{1,2,...,MN\}, bo_{j,o} = v, n \in N \tag{4l}$$

To obtain a feasible solution at the end of iteration, we introduce consumption recipe constraint (4m) for blend components and finished blend-product quality constraint (4n) based on linear blending rules to maintain model linearity.

$$\rho_{s,i}^{c,\min} \sum_{o'\in O,k\in K_p,bo_{j,o'}=v} Lf_{k,o',n} \le \sum_{k\in K_p} Lf_{k,o,n} \le \rho_{s,i}^{c,\max} \sum_{o'\in O,k\in K_p,bo_{j,o'}=v} Lf_{k,o',n},$$
$$\forall j \in J^b, s \in S_j^c, o \in O_s, i \in I_j, ot_{o,i} > 0, v \in \{1,2,...,MN\}, bo_{j,o} = v, n \in N \tag{4m}$$

$$p_{s,p}^{\min} \sum_{o\in O,k\in K_p,ot_{o,i}=1,bo_{j,o}=v} Lf_{k,o,n} \le \sum_{o\in O,k\in K_p,ot_{o,i}=1,bo_{j,o}=v} \left(p_{s,p} Lf_{k,o,n}\right)$$
$$\le p_{s,p}^{\max} \sum_{o\in O,k\in K_p,ot_{o,i}=1,bo_{j,o}=v} Lf_{k,o,n}, \quad \forall j \in J^b, i \in I_j, s \in S_i^p, v \in \{1,2,...,MN\}, n \in N \tag{4n}$$

To restrict the solution space of the model, certain variables relating to blend components outflow from the artificial tanks are fixed to be zero as shown in equation (4o). The PSP must satisfy all the active demand orders ($at_o = 1$) and at least $(MN - 1)$ event points are needed to solve the PSP in order to obtain a feasible solution. We add constraints (4p-4q) in the model to facilitate longer run modes with constant blending rate in blending operations. In equation (4p), PSP is force to fulfill demand order within 3 event-points however this limit can be set higher if needed.

$$Lf_{k,o,n}, l_{k,o,n} = 0, \quad \forall j \in J^b, o \in O^j, k \in K_p, n \in N, n < bo_{j,o} \tag{4o}$$

$$\sum_{k\in K_p,n\in N} l_{k,o,n} \le 3, \quad \forall o \in O, at_o > 0 \tag{4p}$$

$$\sum_{k\in K_p,n\in N} l_{k,o,n} \le MN, \quad \forall o \in O, at_o = 0 \tag{4q}$$

Valid inequalities for PSP are given in (4r-4v). Similar to inequality (1n), constraint (4r) reinforces that if material is flowing out of blend component tank, then it must be being produced by process units at the same event-point. Based on nature of the blend component dummy tanks, we introduce cuts (4s)-(4t) in the formulation of PSP.

$$l_{k,o,n} \leq \sum_{i \in I_s^p, j \in J_i} wv_{i,j,n}, \quad \forall k \in K_p, s \in S_{bc} \cap S_k, o \in O_s, n \in N \tag{4r}$$

$$\sum_{o \in O_s} l_{k,o,n} \leq \sum_{o \in O_s} \sum_{j \in J_k^{pk}} in_{s,j,k,n}, \quad \forall s \in S_{bc}, k \in K_s, n \in N \tag{4s}$$

$$\sum_{j \in J_k^{pk}} \sum_{o \in O_s} l_{k,o,n} \geq \sum_{j \in J_k^{pk}} in_{s,j,k,n}, \quad \forall s \in S_{bc}, k \in K_s, n \in N \tag{4t}$$

Note that the demand order set pertaining to each blend header is arranged in chronologically ascending order. When $bo_{j,o} > bo_{j,o'} > 0$, constraint (4u) states that blend component tanks cannot satisfy demand order o before order o' and equation (4v) restates that if demand order o is satisfied at event-point n then order o' must have been satisfied at earlier even-points.

$$l_{k,o,n'} + l_{k,o',n} \leq 1, \quad \forall k \in K_p, j \in J^b, o \in O, o' \in O, bo_{j,o} > bo_{j,o'}, n > n', n \in N \tag{4u}$$

$$l_{k,o,n} \leq \sum_{k' \in K_s, n' \in N, n' < n} l_{k',o',n'}, \quad \forall k \in K_p, s \in S_{bc} \cap S_k, o \in O_s, o' \in O_s, o \neq o', j \in J^b, b_{j,o} > b_{j,o'}, n \in N \tag{4v}$$

The objective function of the PSP is given in (4w). The second term in the objective function takes into account desire for constant blend rate and the two last terms are necessary to take into consideration the cost incurred if finished blend product demands are not met on time and the cost of storage of finished products in blend scheduling problem. The parameter $tdm = \sum_{s \in S_A \cap S_{bp}, o \in O_s} D_{o,s}^{+}$ is an upper bound on the total amount of finished blend product that can be supplied to the market.

$$\begin{aligned} z_{psp} = & \sum_{i,j\in J_i/J^b,n} c_{i,j}^1 wv_{i,j,n} + \sum_{o\in O,j\in J^b,i\in I_j,bo_{j,o}>0,ot_{o,i}=1,k\in K,n} \frac{c_{i,j}^1 l_{k,o,n}}{\sum_{o'\in O,bo_{j,o'}>0} 1} \\ & + \sum_{s,k\in K_s} c_k^3 st_{s,k,N} + \sum_{j\in J^h/J^b,n} c_j^4 \alpha_{j,n} + \sum_{k\in K^h,n} c_k^5 \beta_{k,n} + \sum_{j\in J^m/J^b,i\in I_j,i'\in I_j,i'\neq i,n} c_{i,i'}^6 \chi_{i,i',j,n} \\ & + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s'} c_{s,s'}^7 \eta o_{s,s',k} + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s',n} c_{s,s'}^8 \eta_{s,s',k,n} \\ & + \sum_{o\in O,j\in J^b,bo_{j,o}>0} \frac{c_o^{14} late_o}{\sum_{o'\in O,bo_{j,o'}>0} 1} + c^{st}\left(\sum_{k\in K,o\in O,n} Lf_{k,o,n} - tdm \right) \end{aligned} \tag{4w}$$

For the purpose of global solution and the algorithm, we report value Z_P calculated from equation (4) using optimal solution of PSP.

$$\begin{aligned} Z^{PSP} = & \sum_{i,j\in J_i,n} c_{i,j}^1 wv_{i,j,n} + \sum_{s,k\in K_s,n} c_k^3 st_{s,k,n} + \sum_{j\in J^h,n} c_j^4 \alpha_{j,n} + \sum_{k\in K^h,n} c_k^5 \beta_{k,n} \\ & + \sum_{j\in J^m,i\in I_j,i'\in I_j,i'\neq i,n} c_{i,i'}^6 \chi_{i,i',j,n} + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s'} c_{s,s'}^7 \eta o_{s,s',k} + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s',n} c_{s,s'}^8 \eta_{s,s',k,n} \end{aligned} \tag{4}$$

6.3.2.1. Acquiring Parameters for RBSP from the Solution of PSP

To solve the restricted blend scheduling problem (RBSP), we determined certain parameters from the optimal solution of the PSP. A new set is defined as $T_j = \{1,2,...,pn_j\}$ where parameter pn_j represents the total number of time periods in which blend components flow into the blend unit j from PSP and $pn_j = \sum_{v\in\{1,2,...,MN\}} \left\{ \sum_{o\in O^j,bo_{j,o}=v,k\in K_p,n\in N} \left(\frac{l_{k,o,n}}{bc^j} \right) \right\}$.

A simple algorithm is used to determine blend components flows into blend units as shown below:

For $j \in J^b$

Initialize $t = 0$

For $n \in N$

If $\sum_{o \in O, k \in K_P, b_{j,o} > 0} l_{k,o,n} \geq 1$

$t = t + 1$

For $s \in S_j^c, o \in O_s, bo_{j,o} > 0, \text{ and } k \in K_s$

If $l_{k,o,n} = 1$

$Uifx_{j,s,t} = Lf_{k,o,n}, Tsx_{j,s,t} = Tos_{k,o,n}, Tfx_{j,s,t} = Tof_{k,o,n}, tx_{j,i,t} = ot_{o,i}$

End Procedure

From the optimal solution of PSP, the information regarding the blend components flow amount ($Uifx_{j,s,t}$) to blend unit *j*, start and finish time ($Tsx_{j,s,t}, Tfx_{j,s,t}$) of these flows, and the task that should be performed ($tx_{j,i,t}$) when blend unit *j* is processing these blend components is obtained.

Set T_j and parameters $Uifx_{j,s,t}, Tsx_{j,s,t}, Tfx_{j,s,t}$ and $tx_{j,i,t}$ are obtained from the optimal solution of PSP and will be used alongside with parameters obtained from BSP in formulating the restricted blend scheduling problem. Formula $\max_j |T_j|$ can be used as a starting point in determining number of event-points needed to obtain a feasible solution of RBSP. If blend scheduling problem can be decomposed into smaller problem than event point is determined separately for each blend scheduling sub-problem. Minimum numbers of event-points for a sub-problem are determined by finding $\max_j |T_j|$ for all blend headers that belong to the sub-problem.

6.3.3. Restricted Blend Scheduling Problem

The restricted blend scheduling problem (RBSP) model incorporates all the constraints of the full-scale model and some additional constraints that are necessary to incorporate the solutions of BSP and PSP. RBSP restricts feasible solution space of BSP by fixing allowable task changeovers for multipurpose blend headers. However unlike BSP, where production in the last event point is not allowed (equation (3d)), RBSP allows blending operations to happen at the last event point to provide flexibility in blending all the material supplied by upstream process units.

The operating mode changeovers at multipurpose blend headers are restricted to those of BSP ($\bar{\chi}_{j,i,i'}$) as shown in equation (5a).

$$\chi_{i,i',j,n} \begin{cases} =1 & if\ \bar{\chi}_{j,i,i'} \neq 0 \\ =0 & if\ \bar{\chi}_{j,i,i'} = 0 \end{cases} \quad , \quad \forall j, i \in I_j, i' \in I_j, i \neq i', n \in N \tag{5a}$$

We introduce a binary variable $x_{j,i,t,n}$ to capture the relationship between supply of blend components to a blend header at time period *t* by PSP and blending of these components at some event point *n* by the blend header. Variable $x_{j,i,t,n}$ is 1 if event point *n* allocates to time period *t* given $tx_{j,i,t}$ is 1.

Constraint (5b) requires that each time period *t* must correspond to at least one event point *n*. Equation (5c) ensures that event point *n* can correspond to at most one time period *t* and equation (5d) forces binary variable $wv_{i,j,n}$ to be 1 if binary variable $x_{j,i,t,n}$ is 1. However, if the event point *n* does not correspond to any time-period *t* then variable $wv_{i,j,n}$ is forced to be zero by constraint (5e).

$$\sum_{n \in N, n \geq t} x_{j,i,t,n} \geq 1, \quad \forall j \in J, i \in I_j, t \in T_j, tx_{j,i,t} = 1 \tag{5b}$$

$$\sum_{t \in T_j, n \geq t, tx_{j,i,t}=1} x_{j,i,t,n} \leq 1, \quad \forall j \in J, i \in I_j, n \in N \tag{5c}$$

$$x_{j,i,t,n} \leq wv_{i,j,n}, \quad \forall j \in J, i \in I_j, n \in N, t \in T_j, n \geq t, tx_{j,i,t} = 1 \tag{5d}$$

$$wv_{i,j,n} \leq \sum_{t \in T_j, n \geq t, tx_{j,i,t}=1} x_{j,i,t,n}, \quad \forall j \in J, i \in I_j, n \in N \tag{5e}$$

An assignment constraint (5f) connects fixed amount of material flowing into a blend unit *j* and time period *t* to an event point n. Continuous variable $UUf_{j,s,t,n}$ represents amount of component *s* supplied by PSP at time period *t* is consumed by a blend header *j* at event point *n*. Constraint (5g) ensures that all of the material supplied to blend header *j* by PSP is blended over scheduling horizon and equation (5h) determines amount of blend component *s* processed by blend unit *j* at event point *n*.

$$UUf_{j,s,t,n} \le Uifx_{j,s,t} x_{j,i,t,n}, \quad \forall j \in J^b, i \in I_j, s \in S_i^c, n \in N, t \in T_j, n \ge t, tx_{j,i,t} = 1 \tag{5f}$$

$$\sum_{\forall n \in N, n \ge t} UUf_{j,s,t,n} = Uifx_{j,s,t}, \quad \forall j \in J^b, i \in I_j, s \in S_i^c, t \in T_j, tx_{j,i,t} = 1 \tag{5g}$$

$$Uif_{s,j,n} = \sum_{t \in T_j, n \ge t, \sum_{i \in I_j} tx_{j,i,t} = 1} UUf_{j,s,t,n}, \quad \forall j \in J^b, s \in S_j^c, n \in N \tag{5h}$$

Equation (5i) sets binary variable $x_{j,i,t,n}$ to zero if flow of materials into blend header *j* at time period *t* is not intended for blending run-mode *i*, that is . $tx_{j,i,t} = 0$. Constraint (5j) fixes amount of material that received and blended by blend header *j* to zero if at given time period *t*, process units are not supplying blend components.

$$x_{j,i,t,n} = 0 \quad if \ tx_{j,i,t} = 0, \quad \forall j^b, i \in I_j, i' \in I_j, i \ne i', n \in N, t \in T_j \tag{5i}$$

$$UUf_{j,s,t,n} = 0 \quad if \ \sum_{i \in I_j} tx_{j,i,t} = 0, \quad \forall j^b, s \in S_j^c, n \in N, t \in T_j \tag{5j}$$

Constraints (5k-5l) determines the duration of each task being performed at unit *j* at event point *n* using component flow rate into blend header at corresponding time period *t*. Equations (5m-5n) bound the start and finish time of each task happening at event point *n* to the start and finish time of the corresponding time period *t*.

$$Uif_{s,j,n} \le \left(\frac{Uifx_{j,s,t}}{Tfx_{j,s,t} - Tsx_{j,s,t}} \right) \left(Tf_{i,j,n} - Ts_{i,j,n} \right) + M\left(2 - wv_{i,j,n} - x_{j,i,t,n} \right), \quad \forall j \in J, i \in I_j, s \in S_i^c, n \in N, t \in T_j, n \ge t, tx_{j,i,t} = 1 \tag{5k}$$

$$Uif_{s,j,n} \ge \left(\frac{Uifx_{j,s,t}}{Tfx_{j,s,t} - Tsx_{j,s,t}} \right) \left(Tf_{i,j,n} - Ts_{i,j,n} \right) - M\left(2 - wv_{i,j,n} - x_{j,i,t,n} \right), \quad \forall j \in J, i \in I_j, s \in S_i^c, n \in N, t \in T_j, n \ge t, tx_{j,i,t} = 1 \tag{5l}$$

$$Ts_{i,j,n} + UH\left(2 - wv_{i,j,n} - x_{j,i,t,n} \right) \ge Tsx_{j,s,t}, \quad \forall j \in J, i \in I_j, n \in N, n \ge t, t \in T_j, tx_{j,i,t} = 1 \tag{5m}$$

$$Tf_{i,j,n} - UH\left(2 - wv_{i,j,n} - x_{j,i,t,n} \right) \le Tfx_{j,s,t}, \quad \forall j \in J, i \in I_j, n \in N, t \in T_j, n \ge t, tx_{j,i,t} = 1 \tag{5n}$$

Constraint (5o) restricts that a time period corresponds to at most three event points similar to constraint (4p) in PSP.

$$\sum_{n \in N, n \ge t} x_{j,i,t,n} \le 3, \quad \forall j \in J, i \in I_j, t \in T_j, tx_{j,i,t} = 1 \tag{5o}$$

To improve the performance of the model, constraint (5p) is added to restrict the feasible space of the model. Constraint (5p) results from the requirement that the scheduling problem must blend components received from PSP. RBSP must satisfy the finished blend product demands hence demand give-away variable $dg_o^l = 0$ and over supply of demand, $dg_o^u \ge 0$, is penalized. Demand demurrages violations are allowed to meet minimum finished blend product demand requirement and these violation are heavily penalized. At least $\max_j \{|T_j|\}$ event points are necessary to obtain a feasible solution of the RBSP.

$$x_{j,i,t,n} = 0, UUf_{j,s,t,n} = 0, \quad \forall j \in J, i \in I_j, s \in S_i^c, tx_{j,i,t} = 1, n \in N, t \in T_j, n < t \tag{5p}$$

The RBSP model has the same objective function as the original full-scale model.

6.3.4. Integer Cuts

If RBSP provides an optimal solution with due-date violations, integer cuts are generated so that they can be added to the BSP in the next iteration. These integer cuts exclude current task changeover sequence for blend headers from future blend scheduling problems. The cuts are produced by deriving alternative combinations of production task assignment to different event points such that the current production task changeover sequence $\bar{\chi}_{i,i',j}$ is maintained. These alternative assignment sequences are feasible solutions of RBSP because RBSP optimizes blend scheduling problem by keeping current task changeover sequence fixed. Table 6.1 shows how alternative combinations of task allocation for blend header are obtained. For example, a BSP that has one multipurpose blend header that can perform 3 tasks to produce 3 different blend products. In the next iteration BSP is to be solved using 5 event-points and current iteration has task changeover sequence for blender (j = 1) obtained from section 6.3.1.1 using BSP solution is $\bar{\chi}_{3,1,1}, \bar{\chi}_{2,3,1}$. Possible alternative solutions of task allocation variable $wv_{i,j,n}$ are provided in Table 6.1 along with the original solution obtained in current iteration. The last event-point has no production taking place because we restrict solution space of BSP as mention in section 6.3.1 by fixing allocation variable $wv_{i,j,N} = 0$. For the given changeover sequence, 6 alternative solutions can be possible and the total of 7 integer cuts are generated as shown in Table 6.1

Hence, multiple cuts can be generated every iteration so that the current changeover sequence is excluded from feasible solution space of BSP in the subsequent iterations of the algorithm. The cuts are only added to those sub-problems that include multi-purpose blend units. Multiple integer cuts of the form (6) are added to the BSP cut-pool for the subsequent iterations.

$$\sum_{(i,j,n)\in Q^{tsc}} wv_{i,j,n} - \sum_{(i,j,n)\in NQ^{tsc}} wv_{i,j,n} \le \left|Q^{tsc}\right| - 1, \quad \forall tsc \in \{1,2,...TSC\} \tag{6}$$

Here, TSC is the total number of solutions that have been obtained to exclude task changeover sequences obtained by current and previous iterations. Set Q^{tsc} includes assignments where $wv_{i,j,n} = 1$ and set NQ^{tsc} includes assignments where $wv_{i,j,n} = 0$. The blend scheduling problem can be decomposed into small independent sub-problems and these independent sub-problems do not share common blend components, products, and product tanks. For instances where more than one sub-problem report due-date violations, cuts are added to one of the sub-problems' cut pool. It is important to note that if BSP, PSP, or RBSP are not globally optimized in steps 1, 2, and 3, adding these integer cuts to the BSP in the next iteration could potentially cut off a better global solution with lower objective value.

Table 6.1 Possible task assignment solutions based on current iteration task changeovers parameter $\bar{\chi}_{i,i',j}$

<table>
<tr><td>BSP solution for multipurpose blend unit</td><td colspan="2">$wv(2,1,n1)$
$wv(2,1,n2)$
$wv(3,1,n3)$
$wv(1,1,n4)$</td><td>Task changeover sequence:
$\bar{\chi}_{2,3,1}$
$\bar{\chi}_{3,1,1}$</td></tr>
<tr><td rowspan="2">Possible solutions</td><td>$wv(2,1,n1)$
$wv(3,1,n2)$
$wv(1,1,n3)$</td><td>$wv(2,1,n1)$
$wv(3,1,n3)$
$wv(1,1,n4)$</td><td>$wv(2,1,n1)$
$wv(3,1,n2)$
$wv(1,1,n4)$</td></tr>
<tr><td>$wv(2,1,n2)$
$wv(3,1,n3)$
$wv(1,1,n4)$</td><td>$wv(2,1,n1)$
$wv(3,1,n2)$
$wv(3,1,n3)$
$wv(1,1,n4)$</td><td>$wv(2,1,n1)$
$wv(3,1,n2)$
$wv(1,1,n3)$
$wv(1,1,n4)$</td></tr>
</table>

6.4. Proposed Heuristic Algorithm

Detailed framework of the algorithm is as follows. The procedure for the algorithm has six steps. We define Zd in equation (7) to be the cost of due-date violations. Z_B^h and Z_P^h are calculated using the optimal objective values of the RBSP and PSP, respectively at the h^{th} iteration.

$$Zd = \sum_{o \in O} c_o^{14} Tearly_o + \sum_{o \in O} c_o^{15} Tlate_o \tag{7}$$

Step 1: Solve the blend scheduling problem (BSP).

Obtain parameters $D_{o,s}^{-}, D_{o,s}^{+}, times_o, timef_o, ot_{o,i}, bo_{j,o}, at_o, tup_o,$ and $\bar{\chi}_{j,i,i'}$ .If infeasible, go to step 6.

Step 2: Based on the requirement of blend components, solve the production unit scheduling problem (PSP) so that the components demand is met with minimum due date violations. The solution of PSP is $Z_P^h = Z^{PSP}$.

Obtained Set T_j and parameters $Uifx_{j,s,t}, Tsx_{j,s,t}, Tfx_{j,s,t}$ and $tx_{j,i,t}$.

Step 3: Restricted BSP (RBSP) is solved using the task changeovers sequence obtained by the solution of BSP for blend units and information about blend components flow into blend units obtained by the solution of PSP. The solution of RBSP is $Z_B^h = Z^{RBSP}$.

Step 4: The global feasible solution is $Z^h = Z_B^h + Z_P^h$.

If the optimal solution of RBSP satisfies the finished blend products demand without any intermediate due-date violations, stop the algorithm and $Z = Z^h$.

If an optimal solution has demurrage violations ($Zd_{RBSP} > 0$), go to step 5. If infeasible, go to step 6.

Step 5: Develop integer cuts based on the task changeover sequence ($\bar{\chi}_{j,i,i'}$) of the BSP solution. Add these cuts to BSP in the next iteration.

$h = h + 1$ and go to step 1.

Step 6: If an optimal solution of RBSP cannot be obtained without demurrage violations or BSP is infeasible, pick the best feasible solution. $Z = \min_h \left(Z^h \right)$ and exit the algorithm.

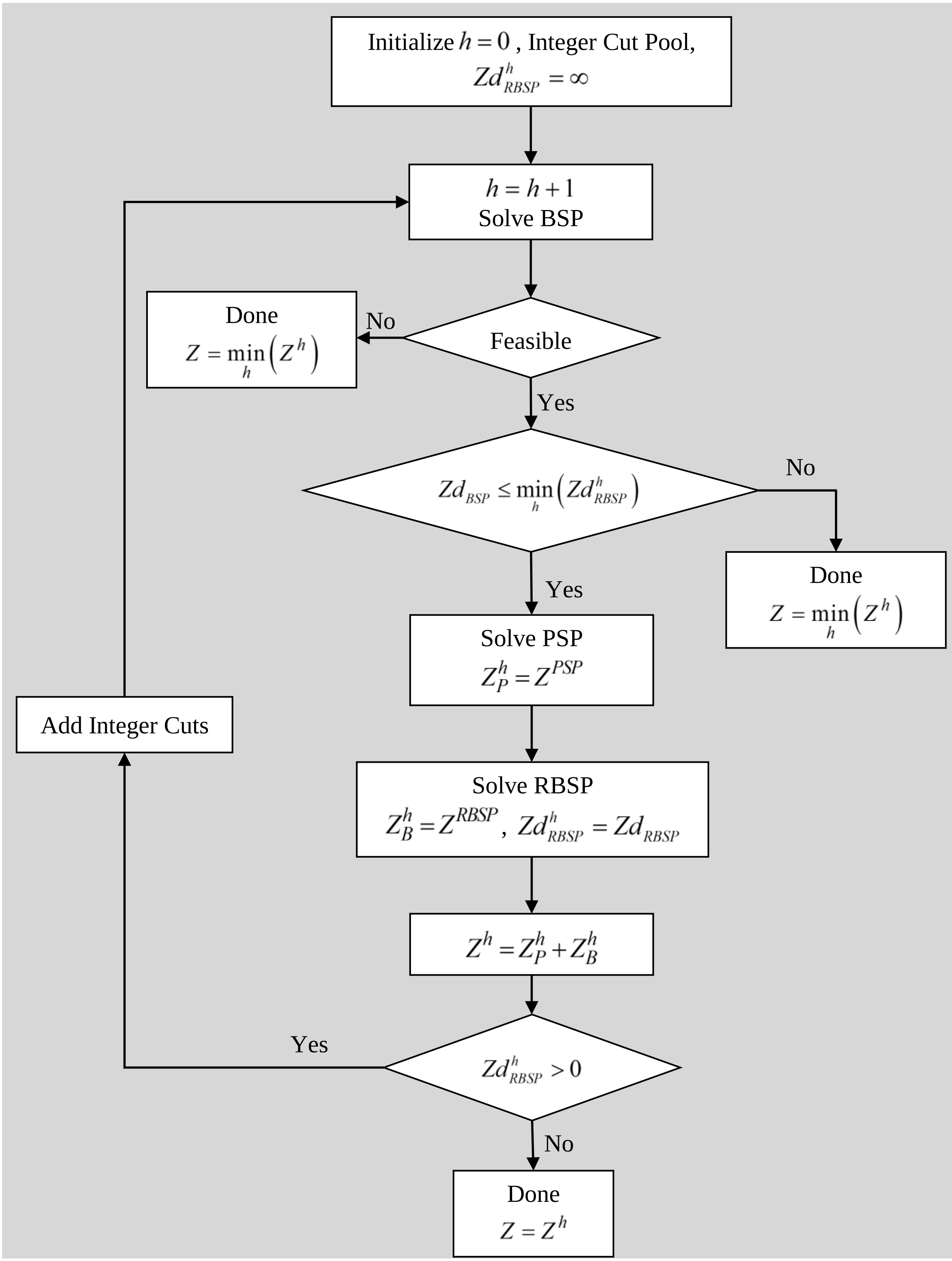

Figure 6.6 Proposed heuristic algorithm

6.4.1. Termination Criterion

The algorithm iterates between solving models BSP, PSP, and RBSP until a global feasible solution with minimum due-date violations and minimum total cost is obtained as presented in Figure 6.6. The optimal schedules of PSP and RBSP provide a feasible solution of the overall refinery problem.

We initialized our BSP model with an empty integer cut-pool. If RBSP reports a solution that respects intermediate demand due dates, then the algorithm terminates and the current solution is kept. If RBSP provides an optimal solution with due-date violations at h^{th} iteration, then corresponding BSP changeover solution ($\bar{\chi}^{h}_{i,i',j}$) is used to generate integer cuts for cut pool. RBSP can never provide a better solution than BSP. Thus, if BSP reports worse solution than the best obtained so far by RBSP, that is $Zd_{BSP} > \min_{h}\left(Zd^{h}_{RBSP}\right)$, then the algorithm is terminated and best global solution obtained so far is kept. BSP can report an infeasible solution when all feasible schedules from its solution space are eliminated due to the addition of integer cuts. In the case of BSP infeasibility, the algorithm terminates and the best solution from previous iterations is reported as a global solution.

At the end of each iteration, the heuristic algorithm provides a feasible solution however the algorithm cannot guarantee a global optimal solution since it terminates when there is no demurrage and not when the overall objective is minimal.

6.4.2. Evaluation of scheduling horizon

The evaluation of BSP, PSP and RBSP problems' scheduling horizon is shown in Figure 6.7. In the Figure 6.7, BSP and RBSP show schedule of a blend unit *j* and PSP shows schedule of materials being supplied to the blend unit ($tx_{j,i,t}$ >0) by process units at event point *n*. A feasible solution of PSP satisfies blend components demands requested by BSP and intermediate products demands (S_B). However due to the production capacity limitation, PSP may not able to meet intermediate due dates of blend components unless PSP utilizes inactive demand orders ($at_o = 0$). In Figure 6.7, the BSP is active only for a total of 4 event points whereas, PSP supplies material to blend unit for 5 event points. The schedule of RBSP is based upon the solution of BSP and PSP; however to meet the due dates of the finished blend products, the time period have flexibility to split. In Figure 6.7, the optimal schedule of RBSP shows that the event point *n3* of PSP is split in two event points, *n3* and *n4* and blend header is performing task with constant blend rate for these event points.

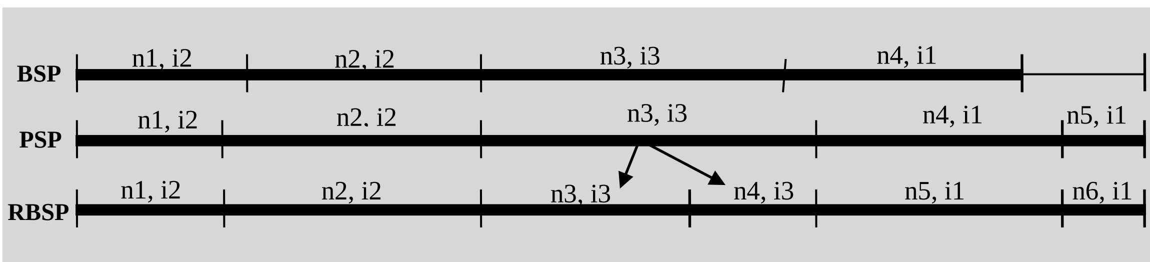

Figure 6.7 Schedule evaluation of BSP, PSP, and RBSP. Here *i* represents index for blend task.

Both PSP and RBSP provide schedules that can be implemented without any further manipulations after algorithm is terminated.

6.5. Case Study

A case study with realistic data provided by Honeywell Process Solutions (HPS) is used to illustrate the effectiveness of the proposed heuristic algorithm. The refinery produces diesel fuels, jet fuel, and other middle distillate products. The production process at Honeywell refinery consists of 2 blender headers, 13 other processing units, and 2 non-identical parallel crude distillation units (CDUs) that process two different types of crude oils. The schematic of production system is shown in Figure 4.2 and details are provided in 4.2.1. Honeywell refinery utilizes rundown blending operations at Jet blender and Diesel blender units to blend components supplied directly from other process units to produce final products. Jet blender unit blends three blend component streams to produce jet fuel that can be stored in 2 product tanks. Diesel blender unit can produces three different grades of fuel: CARB diesel, EPA diesel, and Red-dye diesel by blending three different components in different proportions using three different run-modes called CARB normal, EPA normal, and Red-dye normal, respectively. There are two dedicated tanks for each grade of diesel products and one multipurpose tank that can service CARB and EPA diesel. In addition to Diesel blender, FCC is also a multipurpose unit that has two modes, max distillate mode and max gasoline mode.

To apply the proposed heuristic algorithm, we decompose the case study problem into production unit scheduling and blend scheduling problem. The blend scheduling problem consists of Diesel and Jet blender units and finished product storage tanks associated with these two blenders. The production unit scheduling problem involves all other units and raw material and intermediate storage tanks. The Diesel blender does not consume the same type of blend components as the Jet blender unit and the products produced by these two blenders are serviced by different product tanks, hence, we can decompose the blend scheduling problem (BSP) into two sub-problems, the diesel blend scheduling problem (D-BSP) and the jet blend scheduling problem (J-BSP).

There is a 6 hours of cleaning or maintenance downtime when the multipurpose tank service switches from lower grade of diesel product to higher grade of product. This cleaning downtime is essential to remove any sulfur contamination present in the tank before low sulfur product is sent for storage. The product tank has 4 hours of down time called fill-draw-delay for certificate of analysis preparation and to let the product settle down and mix before is shipped to the market. A minimum run-length restriction of 6 hours is imposed on all production units. The data for this case study for provided in Appendix Chapter 6. Table A6-1 provides maximum and minimum production rate for units, variable production and consumption recipe data for the production units is presented in Table A6-2, and the maximum storage capacity and what material can be stored in each tanks is given in Table A6-3.

6.6. Computational Results

The effectiveness of the proposed heuristic algorithm in solving refinery scheduling problems is demonstrated using ten different examples for the case study presented. Each example differs in either demands, due dates, or initial hold up in the tank. All the examples are solved on a Dell® Optiplex 9020 computer with 2.90 GHz Quad Core Intel® Processor i5-4570S, CPU 2.90GHz, and 16.0 GB memory using CPLEX 12.3.0/GAMS 23.7.1. The heuristic algorithm is implemented in MATLAB R2014a and interfaces with GAMS to solve optimization problems. The maximum solution time of 8 hours and optimality tolerance of 1e-6 are used as termination criteria for all examples. The number of blend product demand orders range from 4 to 22 and with different intermediate due-dates and amount. The scheduling horizon for these problems is 240 hours (10 days). Table 6.2 provides penalty parameters used in objective function and Table A6-4 give demand data for finished blend product with intermediate due-date window. Products CARB diesel, EPA diesel, Red-dye diesel, and Jet fuel are referred to as product P1, P2, P3 and P4, respectively. Intermediate demands are bound by maximum and minimum amount and can be delivered anytime during the delivery window. Maximum delivery rate, product unloading rate to satisfy demand, is 10 (kbbl/h). Demand for distillate product for different examples is given in Table A6-5.

Table 6.2 Penalty parameters in objective function

Penalty Parameter	Value	Penalty Parameter	Value
$C^{1}_{\text{CarbNormal,Diesel Blender}}$	10	$C^{7}_{s,s'}$	40 (Unfavorable: 60)
$C^{1}_{\text{EPA Normal,Diesel Blender}}$	7	$C^{8}_{s,s'}$	40 (unfavorable: 60)
$C^{1}_{\text{Red dye Normal,Diesel Blender}}$	4	C^{9}_{k}	5
$C^{1}_{\text{JetNormal,Jet Blender}}$	5	$C^{10}_{s,k}$	5

C_k^2	1 (Swing tank: 3)	C_o^{11}	1150
C_k^3	$50 \cdot \left(V_k^{\max}\right)^{-1}$	C_o^{12}	800
C_j^4	50	C_s^{13}	1100
C_k^5	10 (Swing tank: 40)	C_o^{14}	5400
$C_{i,i'}^6$	100 (Unfavorable: 250)	C_o^{15}	3250

Model statistics for the proposed algorithm sub-problems and full-scale problems are reported in Table 6.3. For every iteration, the minimum number of event points necessary for optimal solutions are not known prior and need to be determined by trial and error for sub-problems. BSP model is solved using same number of event points during each iteration in the algorithm, as seen in Table 6.3 for examples 3 and 6. Starting point in determining the minimum number of event points for PSP is given in section 6.3.1.1 and the event points for RBSP are determined from optimal solution of PSP as mention in section 6.3.2.1. If PSP optimal solution reports due-date violations for blend component demand orders, then the number of event points necessary for obtaining optimal solution of RBSP may be higher than for corresponding PSP.

Table 6.3 Model statistics

Ex.	**# Orders**	**Full Scale Model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.	**Heuristic Algorithm**				
			BSP model Event pt. Int./Cont. variables (Constraints) Nonzero Elem.		**PSP model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.	**RBSP model** Event pt. Int./Cont. vairables (Constraints) Nonzero Elem.	
			D-BSP	**J-BSP**		**D-RBSP**	**J-RBSP**

1	4	5 360/2663 (6387) 22633	4 97/658 (1698) 6068	2 12/86 (181) 546	4 263/1599 (5412) 17336	5 141/878 (2319) 8536	4 36/382 (663) 1798
2	4	5 360/2663 (6379) 22569	4 97/658 (1692) 6026	2 12/86 (181) 546	4 263/1599 (5412) 17344	4 110/698 (1794) 6324	5 47/482 (844) 2349
3	4	5 360/2663 (6379) 22569	5 124/822 (2169) 8020	2 12/86 (181) 546	5 337/2049 (7381) 23843	5 142/882 (2327) 8526	5 48/486 (861) 2403
			5 124/822 (2170) 8035	2 12/86 (181) 546	5 337/2049 (7381) 23853	5 141/878 (2311) 8472	4 36/382 (663) 1798
			5 124/822 (2171) 8050	2 12/86 (181) 546	5 337/2049 (7381) 23853	5 141/878 (2309) 8472	4 36/382 (663) 1798
			5 124/822 (2178) 8155	2 12/86 (181) 546	5 337/2049 (7381) 23843	5 142/882 (2326) 8526	5 48/486 (861) 2403
			5	2	5	5	4

			124/822 (2185) 8260	12/86 (181) 546	337/2049 (7381) 23853	141/878 (2311) 8472	36/382 (663) 1798
			5 124/822 (2186) 8275	2 12/86 (181) 546	-	-	-
4	6	5 380/2771 (6741) 24018	5 136/886 (2399) 8916	4 32/202 (481) 1516	4 269/1662 (5819) 18733	5 153/942 (2525) 9310	4 42/426 (765) 2160
5	8	6 484/3453 (8644) 31765	6 181/1138 (3227) 12507	4 38/238 (611) 1978	6 426/2610 (10364) 33812	9 322/1858 (5464) 24234	7 98/850 (1686) 5369
6	10	8 680/4737 (12260) 47983	7 226/1386 (4038) 16326	4 44/274 (761) 2524	7 518/3120 (13029) 42828	8 299/1722 (5011) 21059	8 131/1070 (2284) 7794
			7 226/1386 (4087) 17355	4 44/274 (761) 2524	7 518/3120 (13029) 42842	9 341/1946 (5750) 25360	8 131/1070 (2284) 7794
7	13	7 640/4383	7 274/1622	4 44/274	7 509/3120	8 355/1990	8 131/1118

		(11628) 44759	(5057) 20588	(761) 2524	(13349) 43662	(6116) 25900	(2284) 7842
8	16	7 682/4591 (12669) 49601	6 256/1522 (4898) 19356	4 50/310 (931) 3154	6 417/2436 (8739) 28529	7 332/1870 (5916) 24332	7 120/1046 (2219) 7739
9	19	8 841/5509 (16222) 67119	7 328/1886 (6512) 27371	6 100/558 (2061) 7790	7 518/3120 (13029) 42856	8 418/2290 (7722) 33814	9 199/1548 (3929) 15617
10	22	10 1117/7125 (22251) 100257	7 340/1946 (6801) 28643	8 166/870 (3823) 16250	7 512/3021 (12027) 39592	9 493/2658 (9183) 42180	10 261/1914 (5672) 24780

Table 6.4 Computational results

Ex.	**Objective (Z)**							
	Full Scale				**Heuristic Algorithm**			
	PSP	D-BSP	J-BSP	Total	PSP	D-RBSP	J-RBSP	Total
1	109.67	276.21	64.90	450.78	110.88	276.22	60.74	447.84
2	105.16	412.48	81.00	598.64	129.57	421.55	101.46	652.58
3	116.28	14530.12	57.59	14704.00	119.49	15751.44	58.28	15929.21
4	117.66	416.50	58.65	592.81	115.20	427.25	61.11	603.56
5	144.74	294.00	65.46	503.19	150.53	318.00	81.84	550.37

6	154.27	692.85	69.52	916.64	155.67	576.43	80.82	812.92
7	150.54	303.00	59.07	512.61	149.89	309.36	69.26	528.51
8	140.37	446.00	51.00	637.37	156.40	461.15	58.21	675.76
9	165.39	562.46	67.50	795.34	162.48	567.92	71.58	801.99
10	150.69	868.78	79.17	1098.64	166.92	462.31	74.62	703.85

The objective of the proposed algorithm is to obtain a global feasible solution that satisfies minimum demand requirements while minimizing product shipment commitments violations, inventory costs, product giveaway, product tank heel quantity, and undesired mode changeovers at multipurpose blend headers and product tanks. Table 6.4 gives the objective function of full-scale model and heuristic approach. The minimum demand violations are allowed and not fixed to zero in full-scale problem in order to find the optimal solution. To compare quality solution obtained by integrated approach versus decomposition approach, the optimal solution of the integrated problem is used to calculated objective value for each sub-problems. The computational performance results of the full-scale model and the proposed algorithm are shown in Table 6.6. The proposed algorithm is able to obtain a solution at significantly less computational expense than the full-scale model and in some instances full scale model failed to find optimal solutions after 8 hours. For the largest examples with 22 intermediate demand orders, the full-scale has optimality gap of 82.82% after 8 hours whereas the heuristic approach is able to obtain a solution after 865 seconds. For examples 3 and 6, heuristic approach takes more than one iterations before the algorithm terminates and the progress of the algorithm is shown in detail in Table 6.7. Majority of computational effort in the algorithm is spent solving for BSP while RBSP consume less computational time compared to both BSP and PSP. In our work, blend scheduling sub-problems are not solved in parallel however to reduce computational time, they can be solved in parallel. Of two blend scheduling sub-problem, Diesel blend scheduling sub-problem is more complicated due to presence of multipurpose blend header, hence is much harder to solve than Jet blend scheduling sub-problem.

The objective function results obtained using decentralized approach are comparable to that of centralized approach for all examples as shown in Table 6.4. However, for example 1, an optimal solution of integrated full-scale model is higher than that of heuristic approach solution. This happens because our objective function includes a term $\sum_{i,j\in J_i,n} c^1_{i,j} wv_{i,j,n}$. Thus, the objective value of sub-problems is highly dependent on the number of event points used to solve scheduling model. This term is included in objective function as to minimize number of different run modes hence to maintain constant blend rate for longer length of time. In example 1, the full-scale model is solved using 5 event points and diesel and jet blenders are active for all 5 event points, where as in heuristic approach, only diesel blender is active for 5 event points and jet blender for 4 event points since only D-RBSP is solved using 5 points.

$$\begin{aligned} z_{\text{cost}} = & \sum_{k\in K_p,o,n} c^2_k l_{k,o,n} + \sum_{s,k\in K_s} c^3_k st_{s,k,N} + \sum_{j\in J^h,n} c^4_j \alpha_{j,n} + \sum_{k\in K^h,n} c^5_k \beta_{k,n} \\ & + \sum_{j\in J^m,i\in I_j,i'\in I_j,i'\neq i,n} c^6_{i,i'} \chi_{i,i',j,n} + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s'} c^7_{s,s'} \eta o_{s,s',k} + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s',n} c^8_{s,s'} \eta_{s,s',k,n} \\ & + \sum_{k\in K^m,s\in S_k,s'\in S_k,s\neq s',n} c^9_{k\in K^m} std_{s,s',k,n} + \sum_{k\in K^m,s\in S_k,n} c^{10}_{s,k} mh_{s,k,n} + \sum_{o\in O} c^{11}_o dg^l_o + \sum_{o\in O} c^{12}_o dg^u_o \\ & + \sum_{s\in S_f} c^{13}_s rg_s + \sum_{o\in O} c^{14}_o Tearly_o + \sum_{o\in O} c^{15}_o Tlate_o \end{aligned} \tag{8}$$

To effectively compare the quality of the solution between full-scale and heuristic approach, we calculate value Z_{cost} given equation (8) using optimal solution. Here, Z_{cost} is almost same as optimization problem objective function but the term $\sum_{i,j\in J_i,n} c^1_{i,j} wv_{i,j,n}$ is not considered.

When Z_{cost} value is compared for heuristic and full-scale approach, as expected, the full-scale model optimal solution gives lower cost (Z_{cost}) than the proposed algorithm as seen in Table 6.5. Proposed algorithm is successful in obtaining a feasible solution that satisfies minimum and maximum demand requirement while meeting intermediate due-dates for every example for which such a feasible solution exists. Furthermore, heuristic approach is able to provide solution without product giveaway or demand violations for all examples except example 3. For example 3, the full-scale problem reports an optimal solution is with due-date violations and product giveaway. Product giveaway occurs when premium quality product must be given away for the regular product price to meet the regular product demands shipment commitment. The largest example with 22 intermediate demand orders, heuristic provides feasible solution that satisfy demand and intermediate due-dates while minimizing unfavorable blend mode changeovers and product giveaway whereas the full-scale model provides feasible solution with product giveaway and higher unfavorable blend mode changeovers at multipurpose blender.

Table 6.5 Computational results

Ex	**Objective (Z_{cost})**							
	Full Scale				**Heuristic Algorithm**			
	PSP	D-BSP	J-BSP	Total	PSP	D-RBSP	J-RBSP	Total
1	109.67	247.21	39.90	396.78	110.88	247.22	40.74	398.85
2	105.16	384.48	61.00	550.64	129.57	390.55	76.46	596.58
3	116.28	14502.00	37.59	14656.00	119.49	15723.44	38.28	15881.22
4	117.66	388.50	38.65	544.81	115.20	399.25	41.11	555.56
5	144.74	256.00	40.46	440.19	150.23	256.00	46.84	453.37
6	154.27	647.85	39.52	841.64	155.67	519.43	40.82	715.91
7	150.54	261.00	29.07	440.61	149.89	263.36	29.26	442.51
8	140.37	411.00	26.00	577.37	156.40	416.15	28.21	600.76
9	165.39	504.46	32.50	702.34	162.48	509.92	31.58	703.99
10	150.69	809.78	39.17	999.64	166.92	414.31	34.62	615.85

Except for examples 3 and 6, the heuristic algorithm is able to obtain a feasible solution without demurrage in first iteration. The algorithm takes 2 iterations to obtain a global feasible without demurrage in case of the example 6. Even for example 6, heuristic algorithm is able to find a solution in about 564 seconds while full-scale model is not even able to find an optimal solution after 8 hours.

Table 6.6 Computational performance results

Ex.	**Full Scale Model**		**Heuristic Algorithm** CPU time (seconds)					
	CPU time (s)	Gap (%)	D-BSP	J-BSP	PSP	D-RBSP	J-RBSP	Total
1	108.59	0.00	0.44	0.06	2.53	0.20	0.06	3.29
2	249.85	0.00	0.50	1.01	2.76	0.08	0.20	4.56
3	1939.09	0.00	19.11 12.25 12.86 15.38 8.33 11.25	0.047 0.047 0.063 0.062 0.032 0.093	5.32 6.24 6.55 13.10 4.87 -	0.125 0.327 0.250 0.203 0.343 -	0.046 0.031 0.062 0.047 0.062 -	117.09
4	457.27	0.00	1.23	0.047	5.73	0.16	0.031	7.192
5	14716.62	0.00	2.26	0.11	18.31	11.89	0.39	32.96
6	28800	77.98	246.56 141.63	0.047 0.047	90.71 27.32	14.77 40.42	1.72 1.23	564.46
7	28800	9.05	56.52	0.063	28.64	1.37	0.86	87.46
8	28800	69.92	16.51	0.078	33.06	14.62	2.34	66.44
9	28800	74.71	296.73	0.250	37.25	88.41	8.00	430.64

10	28800	82.82	232.10	1.030	51.56	444.87	135.85	865.40

Evolution of the algorithm for example 6 is shown in Table 6.7. In the first iteration, Diesel RBSP obtains optimal solution with intermediate due-date violations. Since only Diesel blend scheduling sub-problem has multi-purpose blender, we produce integer cuts for next iteration using the current iteration task changeover sequence of diesel blender. Diesel blender can perform three tasks (i1, i2, and i3) to produce three different grades of diesel blend products. Changeover sequence ($\bar{\chi}_{i,i',j}$) in first iteration is (i3, i1), (i1, i2) and (i2, i3) and Diesel-BSP sub problem in next iteration is solved using 7 event-points. Total of 49 alternative solutions are obtain from the changeover sequence as described in section 6.3.4 and thus 49 integer cuts are added to D-BSP in 2nd iteration. The algorithm is terminated after 2nd iteration when a global feasible solution without demurrage costs is obtained.

Table 6.7 Algorithm progress and termination

Ex.	**Iter.**	**Objective(Z)** **Demurrage Cost (Zd)**						**Integer Cuts Generated**
		D-BSP	J-BSP	PSP	D-RBSP	J-RBSP	Total	
3	1	2120.64	16.00	107.01	16497.52	63.97	16668.50	1
		886.36	0	0	15247.44	0	15247.44	
	2	2120.64	16.00	119.51	16481.71	55.83	16657.05	1
		886	0	0	15247.44	0	15247.44	
	3	5467.63	16.00	119.49	15751.44	58.28	15929.21	7
		4963.64	0	0	15247.44	0	15247.44	
	4	5467.64	16.00	107.00	15767.25	62.86	15937.11	7
		4963.64	0	0	15247.44	0	15247.44	

	5	8207.00 7150.00	16.00 0	119.50 0	20440.81 19383.81	55.83 0	20616.14 19383.81	1
	6	24006.24 22750.00	16.00 0	-	-	-	-	-
6	1	518.00 0	39.00 0	154.89 0	24852.47 24248.63	79.81 0	25087.17 24248.63	49
	2	518.00 0	39.00 0	155.67 0	576.43 0	80.82 0	812.92 0	-

The heuristic algorithm progress for example 3 is reported in Table 6.7 and the algorithm terminates during 6^{th} iteration and the best solution found so far is kept. For example 3, in the first iteration, D-RBSP sub-problem reports a solution with due date violations thus integer cuts are produced to eliminate the current changeover sequence and are added onto the D-BSP integer cut-pool. In 6^{th} iteration the objective of D-BSP is 24006 and demurrage cost is 22750 which is worse than the best objective of 15929 and demurrage 15247 obtained in 3^{rd} iteration. Thus, the algorithm terminates during 6^{th} iteration because the current D-BSP optimal solution is worse than the best D-RBSP solution obtained so far and RBSP can never do better than its corresponding BSP. The solution obtained by 3^{rd} iteration is reported as the final solution since it has the lowest overall cost of all 5 iterations and lowest demurrage cost. Total of 17 integer cuts are generated at the end of 5^{th} iterations and it takes only about 120 seconds for algorithm to terminate compared to 1939 seconds for full-scale model. For example 3, the full-scale integrated optimal solution reports demurrage violation for a blend product demand order to be 4.32 hours whereas the proposed heuristic approach solution gives violation to be 4.69 hours. Furthermore, the solution obtained using the heuristic algorithm has the same product giveaway cost as full-scale optimal solution.

The performance of our heuristic algorithm is significantly better than that of full-scale model and decomposed problems are able to produce schedules that are similar to that of full-scale model. Figure 6.8 and Figure 6.9 shows Gantt chart for production schedule for example 5 obtain using integrated full-scale model and heuristic model, respectively. The decomposition approach provides similar schedule as integrated approach. Since the refinery under study has blend headers that receive blend components straight from upstream process units, blend headers must always be active when process units are sending materials. This continuous process characteristic of rundown blending is clearly observed in the Gantt charts. It is preferable for blend headers to blend material at constant blend rate. In our heuristic approach, constant blend rate is achieved when material being sent by upstream processes at given time-period is processed during different event-points when restricted blend scheduling problem is solved using production unit scheduling problem results. In Figure 6.9, for the last two event points when Jet blender is using constant blend rate of 0.4826 when blending 106.8 kbbl and 44.5 kbbl amount of jet fuel. Similarly, even though diesel blender has two smaller run modes, blend rate is kept constant between 73.98 to 99.21 hour, and 99.21 to 132.45 hour.

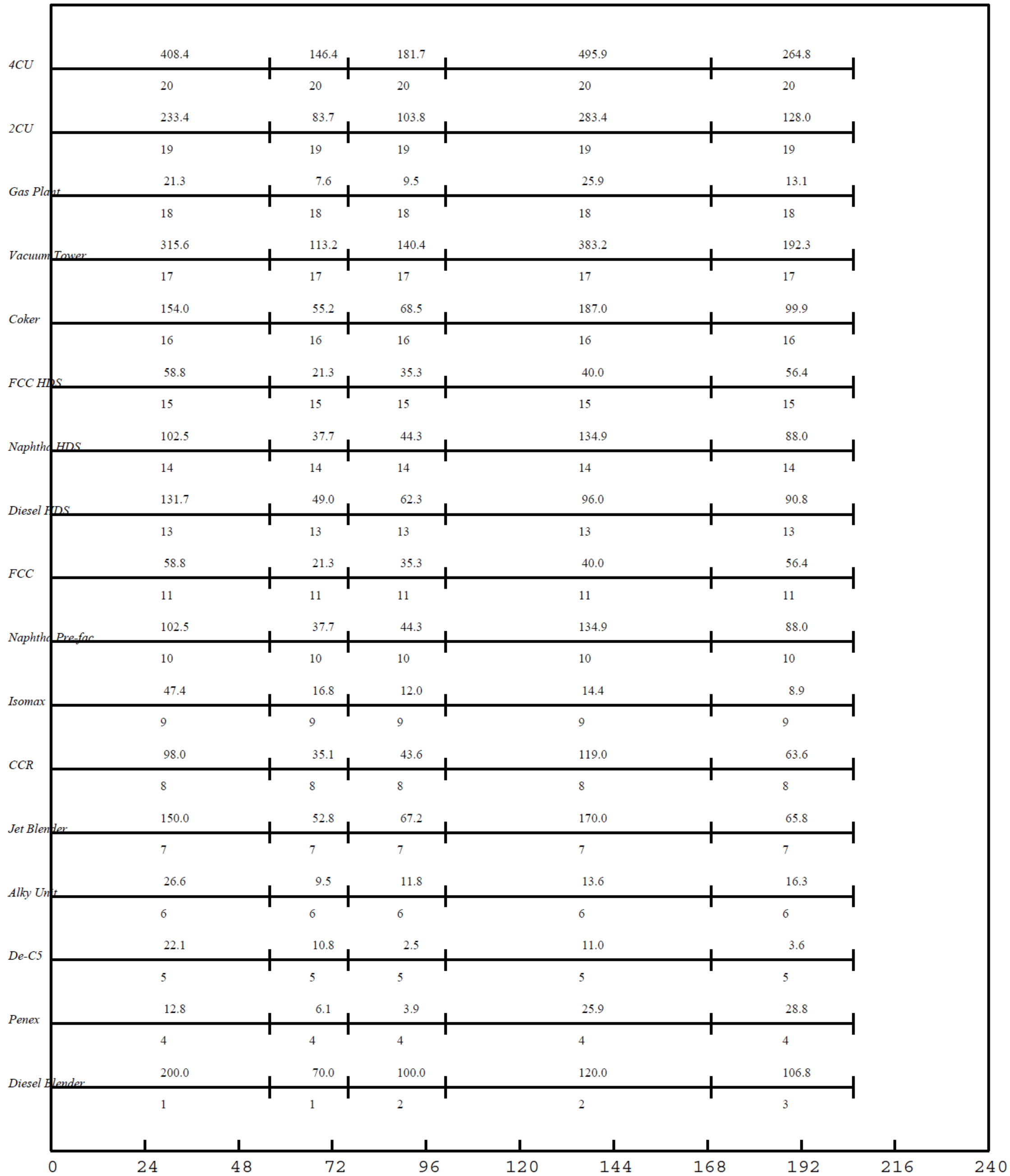

Figure 6.8 Gantt chart for Example 5 obtained using integrated full-scale model

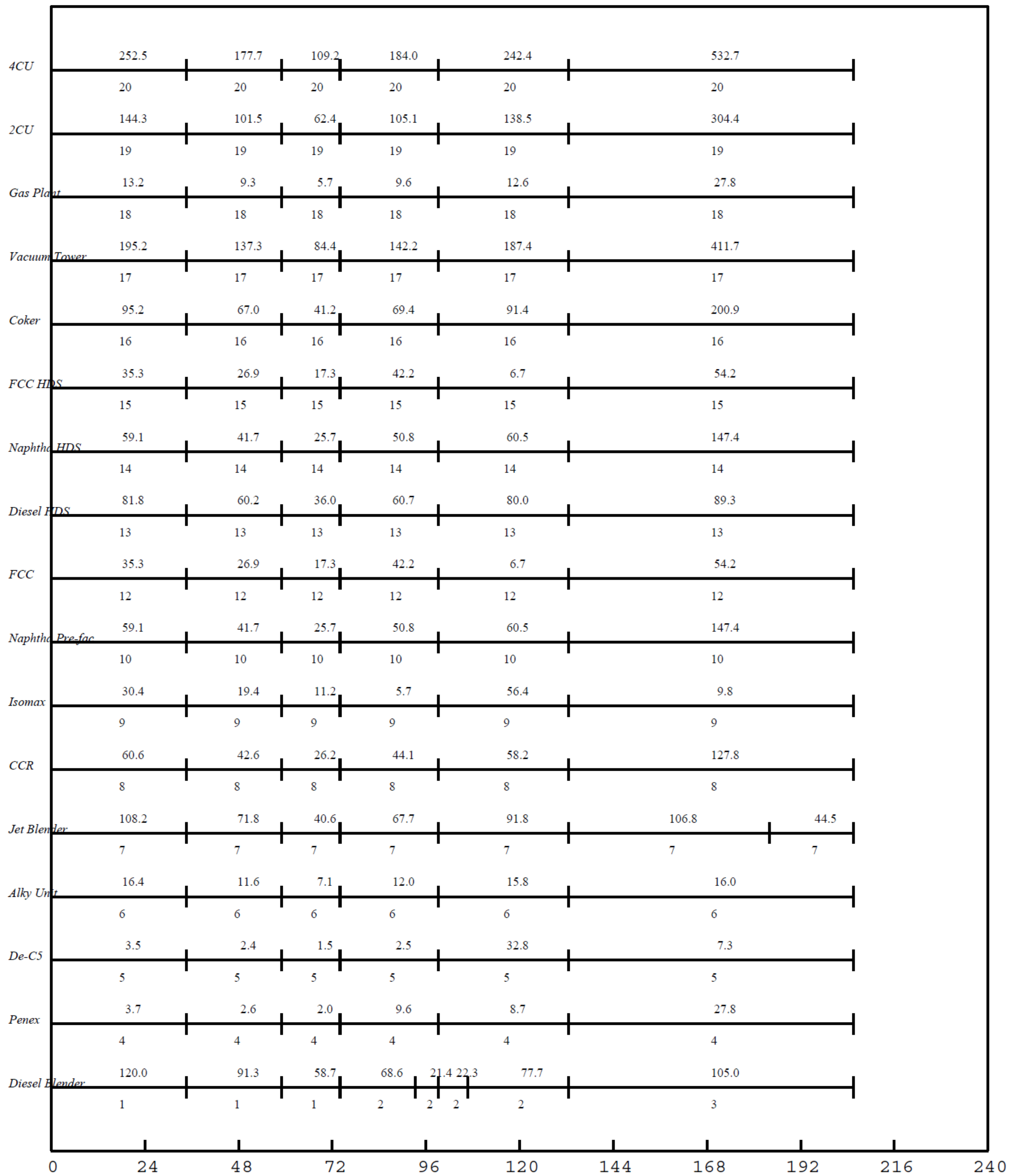

Figure 6.9 Gantt chart for Example 5 obtained from the heuristic algorithm

The work in this chapter is based upon the linear blending rules for finished products and this approach does not reflect the non-linearity of the mixing rules. When the non-linear mixing rules are considered, the scheduling model becomes MINLP instead of MILP which is much harder in terms of computational complexity. The approach proposed here can be utilized as a preprocessing step before tackling complicated MINLP problem. To reduce the complexity of the MINLP scheduling problem, solution space of the non-linear model can be restricted based upon the solution of the pre-processing step. In pre-processing step, the corresponding MILP model of MINLP will be first solved using the heuristic algorithm presented in this work. The obtained feasible solution will provide information on the blend mode changeover sequences for multi-purpose blend headers necessary for meeting minimum demand requirements and intermediate due-dates. By using this information, MINLP feasible solution can be restricted to allow tasks changeovers that are only present in the corresponding MILP. To further reduce complexity of MINLP, other binary and allocation variables can be fixed to the solution obtained from the heuristic algorithm. Proposed algorithm can also be extended to refineries that have blend component storage tanks by making minor changes in PSP model.

6.7. Summary

In this chapter, we present valid inequalities for the large-scale oil-refinery scheduling problems that improve the computational expense necessary to obtain optimal solution. An integrated refinery problem is decomposed into production unit scheduling problem and finished product blending and delivery problem via splitting blend components streams. A heuristic algorithm is proposed based on the decomposed network to obtain a feasible solution of the original problem that satisfies minimum demand requirements while minimizing due-date violations and maximizing performance. It is demonstrated on realistic diesel refinery case study that the proposed algorithm provides feasible solutions that are closer to the optimal solution with significantly less computational effort than that required by the full-scale model.

Nomenclature

Indices

i	Tasks
j	Production units
k	Storage tanks
n	Event points
o	Product order
p	Properties
s	States
t	Periods relating to blend components flow into a unit (calculated from optimal solution of BSP)
v	Periods relating to blend components flow into a unit (calculated from optimal solution of PSP)

Sets

I_j	Tasks which can be performed in unit j
I_s^c / I_s^p	Tasks which can consume/produce material s
J	Production units
J^b	Blend units
J_s^c / J_s^p	Units that consume/produce material s
J_i	Units which are suitable for performing task i
J^h	Units that can produce all the same products as some other unit in the refinery
J^m	Units which are suitable for performing multiple tasks
J_k^{kp} / J_k^{pk}	Units that consume/produce material s stored in tank k
$J_{j'}^{seq}$	Units that follow unit j' (no storage in between)
K	Storage tanks

K^h	tanks that can store the same products as some other tank in the refinery
K_j^{kp} / K_j^{pk}	Tanks that store material consumed/produce by unit j
K^m	Multipurpose tanks that can store multiple materials
K_{bp}	Tanks that can store finished blend products
K_p	Tanks that can store products
K_s	Tanks that can store material s
N	Event point within the time horizon
O	Orders for products that are stored in tanks
P	Product Properties
S	States
S_A	Group A products, stored in tanks
S_B	Group B products, not stored in tanks
S_{bp}	finished blend products, stored in tanks
S_k	Materials that can be stored in tank k
S_{bc}	Blend components
S_{bc}^j	Blend components that can be consumed by blend header j
S_i^c / S_i^p	Materials that can be consumed/produced by task i
S_j^c / S_j^p	Materials that can be consumed/produced by unit j
V_j	Time periods within RBSP during which blend unit j receives component flow

Parameters

at_o	1 if the order o of PSP has to be satisfied by PSP.
$bo_{j,o}$	Order o of PSP is needed by blend unit j

$D_{o,s}^{+}, D_{o,s}^{-}$	Demand limit requirement for order *o* and product *s* that is stored in tank
max_rate_s	Maximum production rate of material *s*
n^j	Number of total event points used by blend unit *j* in BSP model solution
$ot_{o,i}$	1 if task *i* is performed when order *o* of PSP is processed by BSP.
$P_{s,p}$	Property *p* of blend component *s*
$P_{s,p}^{\min} / P_{s,p}^{\max}$	Minimum/maximum specification of property *p* of blend product *s*
pn_j	Total number of time period in which blend components flow into the blend unit *j* from PSP
r_s	Demand of the final product s at the end of the time horizon
$R_{i,j}^{\min} / R_{i,j}^{\max}$	Minimum/ maximum rate of material be processed by task *i* in unit *j*
RL_i	Minimum run length for task *i*
$RU_k^{\min} / RU_k^{\max}$	Minimum/maximum rate of product unloading at tank *k*
$sto_{s,k}$	Amount of state *s* that is present at the beginning of the time horizon in *k*
ta_k	Fill draw delay for product tank *k*
$Tsx_{j,s,t} / Tfx_{j,s,t}$	Start and finish time of blend component *s* flow to blend unit *j* at time period *t*
tup_o	Allowable due-date violations of order *o* without any penalty in PSP
$tx_{j,i,v}$	At time period *v*, blend unit *j* is performing task *i*
UH	Available time horizon
$Uifx_{j,s,v}$	Amount of blend component *s* received by blend unit *j* from PSP at time period *v*
$V_k^{\max}$	Maximum available storage capacity of storage tank *k*
V_k^{heel}	Maximum heel available for storage tank *k*

$yo_{s,k}$	1 if the material *s* is present at the beginning of the time horizon in k
$\alpha_{j,i,i'}$	1 if changeover of task happen from *i* to *i'* at blend unit *j* in BSP
$\rho_{s,i}^{\min}$ / $\rho_{s,i}^{\max}$	Proportion of state *s* produced/consumed by task *i*,

Variables

Binary Variables

$wv_{i,j,n}$	Assignment of task *i* in unit *j* at event point *n*
$in_{s,j,k,n}$	Assigns the material flow of *s* into storage tank *k* from unit *j* at point *n*
$l_{k,o,n}$	Assigns the starting of product flow out of product tank *k* to satisfy order *o* at event point *n*
$out_{s,k,j,n}$	Assigns the material flow of *s* out of storage tank *k* into unit *j* at point *n*
$x_{j,i,v,n}$	Denotes that process material received by blend unit *j* at period *v* is processed by task *i* at event point *n*
$y_{s,k,n}$	Denotes that material *s* is stored in tank k at event point n

Positive variables

$bp_{s,i,j,n}$	Amount of material s produced task *i* in unit *j* at event point *n*
$bc_{s,i,j,n}$	Amount of material s undertaking task *i* in unit *j* at event point *n*
dg_o^l	Minimum demand quantity give-away term for order *o*
dg_o^u	Maximum demand quantity give-away term for order *o*
H	Total time horizon used for production tasks
$JJf_{s,j,j',n}$	Flow of state *s* from unit *j* to consecutive unit *j'* for consumption at point *n*
$Kif_{s,j,k,n}$	Flow of material s from unit *j* to storage tank *k* event point *n*
$Kof_{s,k,j,n}$	Flow of material s from storage tank *k* to unit *j* at point *n*
$late_o$	In PSP, due date violation of blend component demand order *o*

$Lf_{o,k,n}$	Flow of final product for order *o* from storage tank *k* at event point *n*
$mh_{s,k,n}$	Maximum heel give-away term for product tanks
rg_s	Minimum demand quantity give-away term for Group B product *s*
$Rif_{s,k,n}$	Flow of raw material to storage tank *k* event point *n*
$st_{s,k,n}$	Amount of state *s* present in storage tank *k* at event point *n*
$std_{s,s',k,n}$	Amount of state *s* that is downgraded to state s' in storage tank *k* at event point *n*
$Tearly_o$	Early fulfillment of order *o* than required
$Tf_{i,j,n}$	Time that task *i* finishes in unit *j* at event point *n*
$Tlate_o$	Late fulfillment of order *o* than required
$Tos_{k,o,n}$	Time that material starts to flow from tank *k* for order o at event point *n*
$Tof_{k,o,n}$	Time that material finishes to flow from tank *k* for order o at event point *n*
$Ts_{i,j,n}$	Time that task *i* starts in unit *j* at event point *n*
$Tsf_{j,k,n}$	Time that material finishes to flow from unit *j* to tank *k* at event point *n*
$Tsf_{k,j,n}$	Time that material finishes to flow from tank *k* to unit *j* at event point *n*
$Tss_{j,k,n}$	Time that material starts to flow from unit *j* to storage tank *k*
$Tss_{k,j,n}$	Time that material starts to flow from tank *k* to unit *j* at event point *n*
$Uif_{s,j,n}$	Flow of raw material *s* to production unit *j* at point *n*
$Uof_{s,j,n}$	Flow of product material *s* from unit *j* at point *n*
$UUf_{j,s,t,n}$	Amount of state *s* received by unit *j* at period *v* is processed at event *n*
$\alpha_{j,n}$	For unit *j*, 1 if the unit becomes active for very first time at event point *n*
$\beta_{k,n}$	For tank k, 1 if the tank becomes active for very first time at event point *n*

$\eta_{s,s',k,n}$	Continuous 0-1 variable, 1 if material in tank *k* switchover service from s at event point *n* to s' at later event point
$\eta o_{s,s',k}$	Continuous 0-1 variable, 1 if material in tank *k* switchover service from s to s'
$\chi_{i,i',j,n}$	Continuous 0-1 variable, 1 if task at unit *j* changes from *i* at event point *n* to *i'* at later event point.

Chapter 7

7. Augmented Lagrangian Relaxation Approach for Refinery Scheduling

To improve quality of the decision making in the refinery operations, coordination between front-end of a refinery to end-products blending is necessary. Finished product blending and delivery operations and production unit operations present unique challenges each and lead to a complex scheduling model. In this work, we present a unified framework for decomposition via augmented Lagrangian relaxation. The coupling constraints of two operations system are relaxed using augmented Lagrangian relaxation technique and the full-space model is decomposed into production unit scheduling relaxed sub-problem and finished product blending and delivery scheduling relaxed sub-problem. The non-separability terms introduced by augmented penalty are resolved using diagonal quadratic approximation method. A general decomposition algorithm that addresses real-world features such as parallel blend headers, multipurpose product tanks, product giveaway, and blending operations with/without components storage. Applicability of proposed approach is presented on by addressing four different refinery network configurations: no blend component tanks, parallel blend units and no component tanks, component tanks present and no direct flow to blend unit, and direct flow to blend unit and components storage options.

7.1. Introduction

Lagrangian decomposition (LD) method developed in previous chapter addresses scheduling of refinery operations with rundown blending (no blend component tanks) and proposed restricted LD is effective addressing large scale problems. In this chapter, a unified decomposition strategy based on Augmented Lagrangian relaxation method is developed that can address different refinery configurations effectively. In ordinary Lagrangian decomposition, duality gap exists between solution of dual problem and the solution original problem in presence of integer variables or other non-convexities. Thus, heuristic procedures are needed to obtain feasible solution at the end of each iteration. To overcome this problem, an augmented Lagrangian relaxation (ALR) method is adopted. ALR method has been used in several applications (Andreani et al., 2008; Fortin & Glowinski, 1983; Gupta et al., 2001; Z. Li & Ierapetritou, 2010a; Nishi et al., 2008; Tosserams et al., 2006; Tosserams et al., 2008) and strong convergence property of ALR algorithm is proved by (Andreani et al., 2008). One of the drawbacks of ALR method is introduction quadratic term by the augmented penalty term. This nonseparability can be resolved using Diagonal Quadratic Approximation (DQA) method (Ruszczynski, 1995), the Block Coordinate Decent (BCD) method (Bertsekas, 1995; Bertsimas & Sim, 2003), and the Alternating Direction Method (Bertsekas & Tsitsiklis, 1989).

In this chapter, we apply augmented Lagrangian optimization algorithm to decompose refinery operations scheduling problem into production unit scheduling sub-problem and finished product blending and delivery scheduling sub-problem. This chapter is organized as follows. Section 7.2 presents solution strategy, section 7.3 proposes detailed Augmented Lagrangian decomposition (LD) algorithm, and strengthening of decomposition algorithm is presented in section 7.4. Detailed algorithm steps for augmented Lagrangian relaxation are given in section 7.5. Computational results for several different refinery configurations on which algorithm was applied are given in section 7.6 and the chapter concludes with summary in section 7.7.

7.2. Solution Strategy

7.2.1. Augmented Lagrangian method

Augmented Lagrangian relaxation is an appropriate decomposition approach when the problem under study has complicating constraints that upon relaxation leads to smaller sub-problems that are easier to solve. This approach is suitable for integrating two independent scheduling models without needing to obtain full-scale model or using heuristics to find feasible solution. We briefly present a general augmented Lagrangian optimization algorithm for a following linear problem:

Initial Problem (IP)

$$\begin{aligned} z = Min \quad & cx + dy \\ s.t. \quad & Hx - b = 0 \\ & Ax + By \le 0 \\ & x \in \Re_+^n, y \in \Re_+^q \end{aligned}$$

Where, $c \in \Re^n, d \in \Re^q, b \in \Re^m$, A and B are $m \times n$ and $m \times q$ matrices, respectively. To decompose the model (IP) into sub-problems, we introduce duplicating variables xx and coupling constraints $x - xx = 0$.

$$\begin{aligned} z = Min \quad & cx + dy \\ s.t. \quad & Hxx - b = 0 \\ & Ax + By \le 0 \\ & x - xx = 0 \\ & x \in \Re_+^n, xx \in \Re_+^n, y \in \Re_+^q \end{aligned}$$

When coupling constraint are relaxed and incorporated into the objective function. In addition to Lagrangian term, augmented penalty terms are introduced to reduced duality gap.

Relaxed Problem (RP)

$$\begin{aligned} z_R = Min \quad & cx + dy + \lambda^T \left(x - xx\right) + \sigma\left(x - xx\right)^2 \\ s.t. \quad & Hxx - b = 0 \\ & Ax + By \le 0 \\ & x \in \Re_+^n, xx \in \Re_+^n, y \in \Re_+^q \end{aligned}$$

Where, λ represents Lagrangian multipliers and σ is a positive penalty parameters. The augmented penalty term cannot be decomposed as it contains an inseparable cross penalty terms ($x \cdot xx$). To resolve the non-separability issue, diagonal quadratic approximation (DQA) method is used to linearize the cross-product quadratic term around the tentative solution. (Y. Li et al., 2008)

$$(\bar{x}, \overline{xx}).\left(x - xx\right)^2 \approx \left(x - \overline{xx}\right)^2 + \left(\bar{x} - xx\right)^2 - \left(\bar{x} - \overline{xx}\right)^2$$

Thus with above substitution for the non-separable term, the relaxed problem (RP) can be decomposed into sub-problem (RP1) and sub-problem (RP2).

Subproblem Problem (RP1)

$$Z_{R1} = Min \quad cx + dy + \lambda^T x + \sigma\left(x - \overline{xx}\right)^2$$
$$s.t. \quad Ax + By \le 0$$
$$x \in \Re_+^n, y \in \Re_+^q$$

Subproblem Problem (RP2)

$$z_{P2} = Min \quad -\lambda^T xx + \sigma\left\{\left(\bar{x} - xx\right)^2 - \left(\bar{x} - \overline{xx}\right)^2\right\}$$
$$s.t. \quad Hxx - b = 0$$
$$xx \in \Re_+^n$$

Resulting decomposed models (RP1) and (RP2) are quadratic optimization problems due to quadratic terms in the objective function.

The procedure for augmented Lagrangian optimization is as follows:

Step1. Initiate the Lagrangian multiplier $\lambda^{(0)}$ and penalty parameter $\sigma^{(0)} > 0$, $m = 0$

Step 2. For given $\lambda^{(m)}, \sigma^{(m)}, \overline{xx}$, solve sub-problem (RP1). Update $\bar{x} = x$.

Step 3. Solve sub-problem (RP2) based on $\lambda^{(m)}, \sigma^{(m)}$ and $\bar{x}$. Update $\overline{xx} = xx$.

Step 4. Update the Lagrangian multipliers using multiplier method $\lambda^{(m+1)} = \lambda^{(m)} + \sigma^m \left(x - xx\right)^m$.

Step 5. If $\|g\|^m > \gamma \|g\|^{m-1}$, update $\sigma^{m+1} = \beta\sigma^m$. Here, $\gamma \in (0,1)$. For convex objective function, $\beta \ge 1$ is strictly necessary and for fast convergence, $2 < \beta < 3$ range is recommended. (Tosserams et al., 2006)

Step 6. If $\|g\| \le \varepsilon$, the feasible solution is calculated. Here, $\varepsilon > 0$ Otherwise, $m = m + 1$.

Step 7. If iteration number m is larger than the pre-specified number, terminate. Otherwise, go to step 2.

Here, $g = x - xx$, is a coupling constraint violation matrix. Penalty parameter (σ) is updated when the constraint violation is not decreased by a factor γ . Algorithm is terminated when the tolerance is met for constraints violations and feasible solution is found or predefined maximum iteration limit is reached. The augmented Lagrangian optimization algorithm always converges to a feasible solution for $\sigma \to +\infty$.

7.2.2. Decomposition Strategy

Large scale MILPs such as problem (P) proposed in Chapter 4, require specialized solution algorithms that can take into account unique characteristics of unit specific event – points in presence of multi-purpose blend units, and intermediate due-dates. We apply augmented Lagrangian optimization algorithm for solving model (P) and propose modified sub-problems that helps algorithm to converge to an optimal or near optimal solutions. To obtain sub-problems, refinery network is decoupled following the concept of spatial decomposition (e.g. (Karuppiah et al., 2008)). The network is split into two sub-networks, one belonging to production unit operations and other belonging to finished product blending and delivery operations.

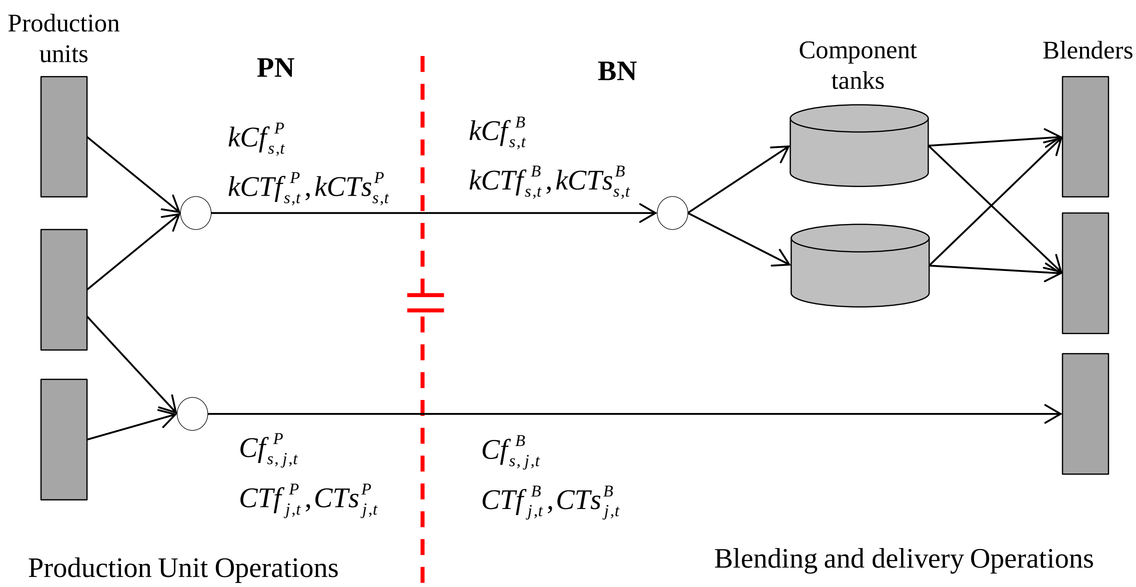

Figure 7.1 Spatial decomposition of a refinery network

To picture physical split, imagine cutting pipelines transferring blend components from production units to blend units or blend component tanks. The variables corresponding to the split pipelines correspond to complicating variables in full-scale model and these variables are amount of material flow and start and finish time of the flow between production operations sub-structure (PN) and blending operations sub-structure (BN). In original model (P), flow of blend component s from production unit j to blend unit j' at event point n ($JJf_{s,j,j',n}$), flow of component s from production unit j to blend component storage tank k at event point n ($Kif_{s,j,k,n}$, $in_{s,j',k,n}$), and start and finish timing variables for flow of material between unit and component tanks ($Tss_{j',k,n}$, $Tsf_{j',k,n}$), are complicating variables. Other variables in model (P) are called non-linking variables since they are separate for both sub-structures PN and BN. Using the approach presented in section 7.2.1, the complicating variables are duplicated, coupling constraints are introduced to model (P) and then these coupling constraints are relaxed in Augmented Lagrangian fashioned using diagonal quadratic approximation method to eliminate non-separable terms. This allows us to decompose the relaxed model (RP) shown below into two models corresponding to sub-networks PN and BN.

$$\begin{aligned}
\min L(\lambda,\mu,\sigma) = z &+ \sum_{s\in S_{bc},j',j,n} \lambda_{s,j',j,n}\left(JJf^{B}_{s,j',j,n} - JJf^{P}_{s,j',j,n}\right) + \sum_{s\in S_{bc},j',k,n} \mu_{s,j',k,n}\left(Kif^{B}_{s,j',k,n} - Kif^{P}_{s,j',k,n}\right) \\
&+ \sum_{j',k\in K_{bc},n} \nu_{j',k,n}\left(Tss^{B}_{j',k,n} - Tss^{P}_{j',k,n}\right) + \sum_{j',k\in K_{bc},n} \upsilon_{j',k,n}\left(Tsf^{B}_{j',k,n} - Tsf^{P}_{j',k,n}\right) \\
&+ \sigma\left\{\sum_{s\in S_{bc},j',j,n}\left(JJf^{B}_{s,j',j,n} - \overline{JJf}^{P}_{s,j',j,n}\right)^2 + \left(\overline{JJf}^{B}_{s,j',j,n} - JJf^{P}_{s,j',j,n}\right)^2 - \left(\overline{JJf}^{B}_{s,j',j,n} - \overline{JJf}^{P}_{s,j',j,n}\right)^2\right\} \\
&+ \sigma\left\{\sum_{s\in S_{bc},j',k,n}\left(Kif^{B}_{s,j',k,n} - \overline{Kif}^{P}_{s,j',k,n}\right)^2 + \left(\overline{Kif}^{B}_{s,j',k,n} - Kif^{P}_{s,j',k,n}\right)^2 - \left(\overline{Kif}^{B}_{s,j',k,n} - \overline{Kif}^{P}_{s,j',k,n}\right)^2\right\} \\
&+ \sigma\left\{\sum_{j',k\in K_{bc},n}\left(Tss^{B}_{j',k,n} - \overline{Tss}^{P}_{j',k,n}\right)^2 + \left(\overline{Tss}^{B}_{j',k,n} - Tss^{P}_{j',k,n}\right)^2 - \left(\overline{Tss}^{B}_{j',k,n} - \overline{Tss}^{P}_{j',k,n}\right)^2\right\} \\
&+ \sigma\left\{\sum_{j',k\in K_{bc},n}\left(Tsf^{B}_{j',k,n} - \overline{Tsf}^{P}_{j',k,n}\right)^2 + \left(\overline{Tsf}^{B}_{j',k,n} - Tsf^{P}_{j',k,n}\right)^2 - \left(\overline{Tsf}^{B}_{j',k,n} - \overline{Tsf}^{P}_{j',k,n}\right)^2\right\}
\end{aligned}$$

s.t. constraints corresponding and variables in (P) proposed in Chapter 4 (RP)

However, to avoid introducing large set of complicating variables ($JJf_{s,j,j',n}$, $Kif_{s,j,k,n}$, $in_{s,j',k,n}$, $Tss_{j',k,n}$, $Tsf_{j',k,n}$), we aggregate the blend component s streams from multiple production units supplying BN into one stream supplying BN as seen in Figure 7.1. Similarly, multiple inlet streams for a blend component s that has storage option available, is aggregated into one inlet stream which is then split into multiple stream connecting different component tanks (Figure 7.1). New the complicating variables are reduced to ($Cf_{s,j,n}$, $KCf_{s,n}$, $KCTs_{s,n}$, $KCTf_{s,n}$). Here, $Cf_{s,j,n}$ is flow of component s from sub-structure PN into blend unit j in BN an event-point n. For blend component that can be stored in component tanks, $KCfn_{s,n}$ is flow of component s from PN to BN at event-point n, and $KCTs_{s,n} / KCTf_{s,n}$ is start/finish time of flow of component s from PN to BN at event-point n.

Complicating variables are in terms of continuous time representation and notion of unit specific event points. When augmented Lagrangian algorithm is implemented, this types of model leads to sub-optimal solutions and oscillations in objective function in the beginning of the algorithm, especially for a refinery configuration that has no blend component storage options and multipurpose blend units. To eliminate this behavior, we introduce a new transfer event-point t that is only active if there is flow of material between production unit operations and blending operations and the complicating variables can we replaced with ($Cf_{s,j,t}$, $KCf_{s,t}$, $KCTs_{s,t}$, $KCTf_{s,t}$). The detail information regarding new transfer event-point t and its relationship with event-point n is provided in detail in section 7.3. Furthermore, the start and finish time of component s flow between PN and a blend unit j are not defined explicitly in full-scale model and therefore these complicating variables ($CTs_{j,t}$, $CTf_{j,t}$) need to be defined in decomposed models. The timing variables for a component s flow between PN and a blend unit j at transfer event- point t is defined as $CTs_{j,t}$ and $CTf_{j,t}$ instead of $CTs_{s,j,t}$ and $CTf_{s,j,t}$ because all components that are supplied directly to the blend unit must have same start and finish times due to continuous production.

Using the updated set of complicating variables ($Cf_{s,j,t}$, $KCf_{s,t}$, $CTs_{j,t}$, $CTf_{j,t}$, $KCTs_{s,t}$, $KCTf_{s,t}$), we obtain a new relaxed problem (LRP) where old complicating variables ($JJf_{s,j,j',n}$, $Kif_{s,j,k,n}$, $in_{s,j',k,n}$, $Tss_{j',k,n}$, $Tsf_{j',k,n}$) connecting two sub-structures and related constraints are eliminated from model (RP).

$$
\begin{aligned}
\min\ f = z &+ \sum_{s\in S_{bc},\, j\in J^{BU},\, t\in T} \lambda_{s,j,t}\left(Cf^{B}_{s,j,t} - Cf^{P}_{s,j,t}\right) + \sum_{j\in J^{BU},\, t}\left\{+\pi_{j,t}\left(CTs^{B}_{j,t} - CTs^{P}_{j,t}\right) + \tau_{j,t}\left(CTf^{B}_{j,t} - CTf^{P}_{j,t}\right)\right\} \\
&+ \sum_{s\in S_{bc},\, t}\left\{\mu_{s,t}\left(KCf^{B}_{s,t} - KCf^{P}_{s,t}\right) + \nu_{s,t}\left(KCTs^{B}_{s,t} - KCTs^{P}_{s,t}\right) + \upsilon_{s,t}\left(KCTf^{B}_{s,t} - KCTf^{P}_{s,t}\right)\right\} \\
&+ \sigma \sum_{s\in S_{bc},\, j\in J^{BU},\, t}\left\{\left(Cf^{B}_{s,j,t} - \overline{Cf}^{P}_{s,j,t}\right)^2 + \left(\overline{Cf}^{B}_{s,j,t} - Cf^{P}_{s,j,t}\right)^2 - \left(\overline{Cf}^{B}_{s,j,t} - \overline{Cf}^{P}_{s,j,t}\right)^2\right\} \\
&+ \sigma \sum_{j\in J^{BU},\, t}\left\{\begin{aligned}&\left(CTs^{B}_{j,t} - \overline{CTs}^{P}_{j,t}\right)^2 + \left(\overline{CTs}^{B}_{j,t} - CTs^{P}_{j,t}\right)^2 - \left(\overline{CTs}^{B}_{j,t} - \overline{CTs}^{P}_{j,t}\right)^2 \\ &+\left(CTf^{B}_{j,t} - \overline{CTf}^{P}_{j,t}\right)^2 + \left(\overline{CTf}^{B}_{j,t} - CTf^{P}_{j,t}\right)^2 - \left(\overline{CTf}^{B}_{j,t} - \overline{CTf}^{P}_{j,t}\right)^2\end{aligned}\right\} \\
&+ \sigma \sum_{s\in S_{bc},\, t}\left\{\begin{aligned}&\left(KCf^{B}_{s,t} - \overline{KCf}^{P}_{s,t}\right)^2 + \left(\overline{KCf}^{B}_{s,t} - KCf^{P}_{s,t}\right)^2 - \left(\overline{KCf}^{B}_{s,t} - \overline{KCf}^{P}_{s,t}\right)^2 \\ &+\left(KCTs^{B}_{s,t} - \overline{KCTs}^{P}_{s,t}\right)^2 + \left(\overline{KCTs}^{B}_{s,t} - KCTs^{P}_{s,t}\right)^2 - \left(\overline{KCTs}^{B}_{s,t} - \overline{KCTs}^{P}_{s,t}\right)^2 \\ &+\left(KCTf^{B}_{s,t} - \overline{KCTf}^{P}_{s,t}\right)^2 + \left(\overline{KCTf}^{B}_{s,t} - KCTf^{P}_{s,t}\right)^2 - \left(\overline{KCTf}^{B}_{s,t} - \overline{KCTf}^{P}_{s,t}\right)^2\end{aligned}\right\}
\end{aligned}
$$

s.t. constraints corresponding to non-linking variables in (P) (LRP)

Following the augmented Lagrangian decomposition approach presented in section 7.2.1, model (LRP) can be decomposed into two relaxed sub-problems (LBSP) and (LPSP) such that model (LBSP) includes equations and variables pertaining to sub-network BN, while model (LPSP) includes equations and variables corresponding to the structure PN. The bounds of non-linking variables in both the both sub-problems are same as original full-space model (P) whereas the original linking variables are eliminated from the original full-space problem (P) and will be replaced with new variables that will be defined in section 7.3. The two models (LBSP) and (LPSP) are as follows:

$$
\begin{aligned}
\min\ f^{BSP} = z &+ \sum_{s\in S_{bc},\, j\in J^{BU},\, t} \lambda_{s,j,t} Cf^{B}_{s,j,t} + \sum_{j\in J^{BU},\, t}\left(\pi_{j,t} CTs^{B}_{j,t} + \tau_{j,t} CTf^{B}_{j,t}\right) \\
&+ \sum_{s\in S_{bc},\, t}\left(\mu_{s,t} KCf^{B}_{s,t} + \nu_{s,t} KCTs^{B}_{s,t} + \upsilon_{s,t} KCTf^{B}_{s,t}\right) \\
&+ \sigma \sum_{s\in S_{bc},\, j\in J^{BU},\, t}\left(Cf^{B}_{s,j,t} - \overline{Cf}^{P}_{s,j,t}\right)^2 + \sigma \sum_{j\in J^{BU},\, t}\left\{\left(CTs^{B}_{j,t} - \overline{CTs}^{P}_{j,t}\right)^2 + \left(CTf^{B}_{j,t} - \overline{CTf}^{P}_{j,t}\right)^2\right\} \\
&+ \sigma \sum_{s\in S_{bc},\, t}\left\{\left(KCf^{B}_{s,t} - \overline{KCf}^{P}_{s,t}\right)^2 + \left(KCTs^{B}_{s,t} - \overline{KCTs}^{P}_{s,t}\right)^2 + \left(KCTf^{B}_{s,t} - \overline{KCTf}^{P}_{s,t}\right)^2\right\}
\end{aligned}
$$

s.t. constraints corresponding to units and connections in BN (LBSP)

$$\begin{aligned}
\min\ f^{PSP} = z &- \sum_{s\in S_{bc},\, j\in J^{BU},\, t}\left(\lambda_{s,j,t} Cf^{P}_{s,j,t}\right) - \sum_{j\in J^{BU},\, t}\left(\pi_{j,t} CTs^{P}_{j,t} + \tau_{j,t} CTf^{P}_{j,t}\right) \\
&- \sum_{s\in S_{bc},\, t}\left(\mu_{s,t} KCf^{P}_{s,t} + \nu_{s,t} KCTs^{P}_{s,t} + \upsilon_{s,t} KCTf^{P}_{s,t}\right) \\
&+ \sigma \sum_{s\in S_{bc},\, j\in J^{BU},\, t}\left\{\left(\overline{Cf}^{B}_{s,j,t} - Cf^{P}_{s,j,t}\right)^2 - \left(\overline{Cf}^{B}_{s,j,t} - \overline{Cf}^{P}_{s,j,t}\right)^2\right\} \\
&+ \sigma \sum_{j\in J^{BU},\, t}\left\{\begin{aligned}&\left(\overline{CTs}^{B}_{j,t} - CTs^{P}_{j,t}\right)^2 - \left(\overline{CTs}^{B}_{j,t} - \overline{CTs}^{P}_{j,t}\right)^2 \\ &+\left(\overline{CTf}^{B}_{j,t} - CTf^{P}_{j,t}\right)^2 - \left(\overline{CTf}^{B}_{j,t} - \overline{CTf}^{P}_{j,t}\right)^2\end{aligned}\right\} \\
&+ \sigma \sum_{s\in S_{bc},\, t}\left\{\begin{aligned}&\left(\overline{KCf}^{B}_{s,t} - KCf^{P}_{s,t}\right)^2 - \left(\overline{Kif}^{B}_{s,j',k,n} - \overline{Kif}^{P}_{s,j',k,n}\right)^2 \\ &+\left(\overline{KCTs}^{B}_{s,t} - KCTs^{P}_{s,t}\right)^2 - \left(\overline{KCTs}^{B}_{s,t} - \overline{KCTs}^{P}_{s,t}\right)^2 \\ &+\left(\overline{KCTf}^{B}_{s,t} - KCTf^{P}_{s,t}\right)^2 - \left(\overline{KCTf}^{B}_{s,t} - \overline{KCTf}^{P}_{s,t}\right)^2\end{aligned}\right\}
\end{aligned}$$

s.t. constraints corresponding to units and connections in PN (LPSP)

Here, $\lambda_{s,j,t}$, $\pi_{s,j,t}$, $\tau_{s,j,t}$, $\mu_{s,t}$, $\nu_{s,t}$, and $\upsilon_{s,t}$ represent Lagrangian multipliers and σ is the corresponds to augmented Lagrangian penalty term. The scheduling model (LBSP) and (LPSP) can be solved using different number of event points, $ep^{\min}_{PN}$ and $ep^{\min}_{BN}$ corresponds to the minimum number of event-points (n) required to obtain optimal solution of (LPSP) and (LBSP), respectively. The cardinality of transfer event-point set is equal to $ep^{\min}_{PN}$, that is set $T = \left\{1, 2, \ldots, ep^{\min}_{PN}\right\}$.

7.3. Decomposed Models

In addition to non-linking variables and constraints present in (P), decomposed models (LBSP) and (LPSP) includes additional constraints and variables to define complicating variables in terms of new transfer event-point t. We first present finished product blending and delivery scheduling problem (LBSP) and then relaxed production unit scheduling model (LPSP).

7.3.1. Relaxed Blend Scheduling Problem

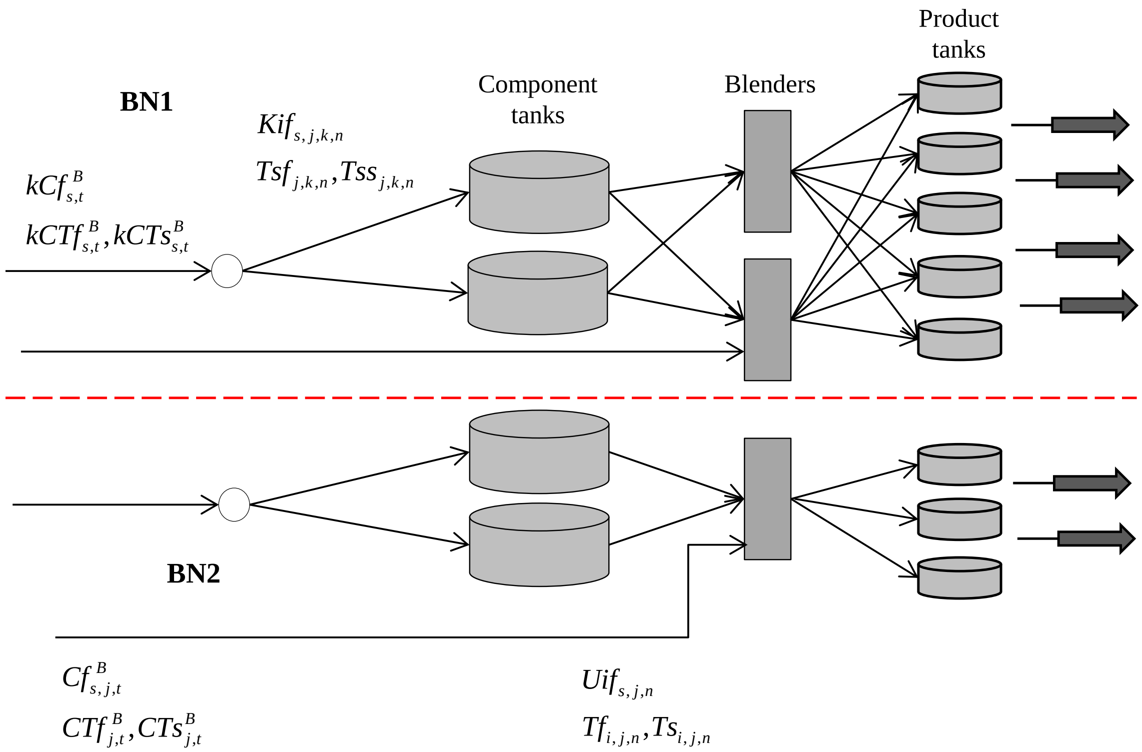

Figure 7.2 Spatial decomposition of blend sub-network structure (BN)

Finished product blending and delivery operations correspond to sub-structure BN shown in Figure 7.2 in detail. In this problem we assume that all blend component tanks are dedicated tank and blend components are either supplied directly to a blend unit or are supplied to component tanks. Blending operations features parallel blend units, multiple non-identical blend units producing different products, multipurpose blend units, and multipurpose product tanks. BN can be decomposed into smaller sub-structures using following rule: sub-structures do not have common blend components, products, units, and tanks. Spatial decomposition using this rule is carried out in Figure 7.2 and we obtain two sub-structures BN1 and BN2. Based on this decomposition, problem (LBSP) can be decompose into two independent problems (LBSP1) and (LBSP2), where model (LBSP1) contain all the variables and equations relating to sub-network BN1 and problem (LBSP2) includes all the variables and equations relating to sub-network BN2. For a given refinery network, additional decomposition of BN can be carried out to obtain more than 2 sub-networks and corresponding decomposed scheduling models if possible and set BN includes all the decomposed sub-network of BN, e.g. $BN = \{BN1, BN2\}$. These independent decomposed scheduling models can be solved using different number of event-points and parameter $ep_{bn}^{\min}$ gives minimum number of event-points needed to solve scheduling problem corresponding to sub-network bn to optimality.

Since original complicating variables ($JJf_{s,j,j',n}$, $Kif_{s,j,k,n}$, $in_{s,j',k,n}$, $Tss_{j',k,n}$, $Tsf_{j',k,n}$) are eliminated from (LBSP), we make following changes:

- If a blend component is supplied directly to unit without storage options, then the blend component is treated as a raw material for blend unit and flow of component s to blend unit j at event-point n is given by $Uif_{s,j,n}$
- A unit is added set J for BN to represent a dummy production unit
- For blend component that can be stored into tanks, variable $Kif_{s,j,k,n}$ represent flow of component s into tank k from dummy production unit j at event-point n

The start and finish time ($Tss_{j,k,n}$, $Tsf_{j,k,n}$) for flow into the component tank at event-point n is define by Eq. (1).

$$Kif_{s,j,k,n} \le M\left(Tsf_{j,k,n} - Tss_{j,k,n}\right) + M\left(1 - in_{s,j,k,n}\right), \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N \tag{1}$$

Here, the binary variable $in_{s,j,k,n}$ is defined by equation (7a) in Chapter 4 and is equal to 1 if the blend component is flowing into tank at event-point n. If component is supplied to the tank from a dummy production unit j at event-point n, then total amount is bounded by product of big-M and duration of flow into the tank.

To represent complicating variables in terms of transfer event-point t instead of event-point n, we introduce 0-1 continuous variables $x_{j,n,t}$ and $xk_{s,n,t}$. These variables are 1 if a transfer event-point t to corresponds to an active event-point n, otherwise 0.

$$x^{B}_{j,n,t} \le \sum_{i \in I_j} wv_{i,j,n}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{2a}$$

$$x^{B}_{j,n,t} \le 1 - \sum_{t' < t, t' \le n} x^{B}_{j,n,t'}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{2b}$$

$$x^{B}_{j,n,t} \le 1 - \sum_{n' < n, n' \ge t} x^{B}_{j,n',t}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{2c}$$

$$x^{B}_{j,n,t} \ge \sum_{i \in I_j} wv_{i,j,n} - \sum_{t' < t, t' \le n} x^{B}_{j,n,t'} - \sum_{n' < n, n' \ge t} x^{B}_{j,n,t'}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{2d}$$

$$\sum_{n \ge t'} x^{B}_{j,n,t'} \le \sum_{n \ge t} x^{B}_{j,n,t}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T, t' \in T, t' > t \tag{2e}$$

$$x^{B}_{j,n,t} + x^{B}_{j,n',t'} \le 1, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, n' \in N, n < n', t \in T, t' \in T, t' < t, n \ge t, n' \ge t, n' \ge t', n \ge t' \tag{2f}$$

$$x^{B}_{j,n,t} = 0, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n < t \tag{2g}$$

Constraints (2a)-(2g) are for a blend unit that receives at least one blend component directly from production unit operations. Binary assignment variable $wv_{i,j,n}$ is 1 if task i is active at event-point n for unit j. For a blend unit j that is performing some task at event-point n, Eqs. (2a)-(2d) only one event-point t to the event-point n. Eqs. (2e) enforce that event-point $t' < t$ should be assign an active event-point n before event-point t. More ever, for any two active event-points $n' < n$, n' must be assigned to an event-point t' before assigning n to an event-point t, where $t' < t$. Constraints (2a)-(2e) eliminates an occurrence of a situation where $x_{j,n,t} = 1$ and $x_{j,n,t'} = 0$ for $t' < t$. Since there is one to one correspondence between an active event-point n and event-point t for a unit j, we introduce Eq. (2g) to fix variable $x_{j,n,t} = 0$ whenever $n < t$.

Similar to constraints (2a)-(2g) for blend unit that receive components directly from production unit operations, we present Eqs. (3a)-(3g) for blend component tanks as follows:

$$xk^{B}_{s,n,t} \le \sum_{k \in K_s, j \in J_{dmy}} in_{s,j,k,n}, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{3a}$$

$$xk^{B}_{s,n,t} \le 1 - \sum_{t' < t, t' \le n} xk^{B}_{s,n,t'}, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{3b}$$

$$xk_{s,n,t}^{B} \le 1 - \sum_{n'<n,n'\ge t} xk_{s,n',t}^{B}, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{3c}$$

$$xk_{s,n,t}^{B} \ge \sum_{j \in J_{dmy}} in_{s,j,k,n} - \sum_{t'<t,t'\le n} xk_{s,n,t'}^{B} - \sum_{n'<n,n'\ge t} xk_{s,n,t'}^{B}, \quad \forall s \in S_{bc} \cap S_k, k \in K_s, n \in N, t \in T, n \ge t \tag{3d}$$

$$\sum_{n \ge t'} xk_{s,n,t'}^{B} \le \sum_{n \ge t} xk_{s,n,t}^{B}, \quad \forall s \in S_{bc} \cap S_k, t \in T, t' \in T, t' > t \tag{3e}$$

$$xk_{s,n,t}^{B} + xk_{s,n',t'}^{B} \le 1, \quad \forall s \in S_{bc} \cap S_k, n \in N, n' \in N, n < n', t \in T, t' \in T, t' < t, n \ge t, n' \ge t, n' \ge t', n \ge t' \tag{3f}$$

$$xk_{s,n,t}^{B} = 0, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n < t \tag{3g}$$

Complicating variables ($Cf_{s,j,t}$, $KCf_{s,t}$, $CTs_{s,j,t}$, $CTf_{s,j,t}$, $KCTs_{s,t}$, $KCTf_{s,t}$) expressed through transfer event-point t are defined below:

$$Cf_{s,j,t}^{B} \le Uif_{s,j,n} + M\left(1 - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, n \in N, t \in T, n \ge t \tag{4a}$$

$$Cf_{s,j,t}^{B} \ge Uif_{s,j,n} - M\left(1 - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, n \in N, t \in T, n \ge t \tag{4b}$$

$$Cf_{s,j,t}^{B} \le M \sum_{n \ge t} x_{j,n,t}^{B}, \quad \forall j \in J^{BU}, s \in S_j^c / S_k, t \in T \tag{4c}$$

$$CTs_{j,t}^{B} \le Ts_{i,j,n} + UH\left(2 - wv_{i,j,n} - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU} \cap J^{ds}, i \in I_j, n \in N, t \in T, n \ge t \tag{5a}$$

$$CTs_{j,t}^{B} \ge Ts_{i,j,n} - UH\left(2 - wv_{i,j,n} - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU} \cap J^{ds}, i \in I_j, n \in N, t \in T, n \ge t \tag{5b}$$

$$CTf_{j,t}^{B} \le Tf_{i,j,n} + UH\left(2 - wv_{i,j,n} - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU} \cap J^{ds}, i \in I_j, n \in N, t \in T, n \ge t \tag{5c}$$

$$CTf_{j,t}^{B} \ge Tf_{i,j,n} - UH\left(2 - wv_{i,j,n} - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU} \cap J^{ds}, i \in I_j, n \in N, t \in T, n \ge t \tag{5d}$$

$$CTs_{j,t}^{B} \le UH \sum_{n \ge t} x_{j,n,t}^{B}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T \tag{5e}$$

$$CTf_{j,t}^{B} \le UH \sum_{n \ge t} x_{j,n,t}^{B}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T \tag{5f}$$

Flow of component s from sub-network PN to blend unit j at transfer event-point t is equal to raw material flow into the blend unit at event-point n if $x_{j,n,t}$ equal to 1 as stated by Eqs. (4a)-(4b). Eq. (50c) forces $Cf_{s,j,t}$ to be zero if at transfer event point t, no flow exists between PN and a blend unit in BN. As given in Eqs. (5a)-(5d), timing variables for component flows at event-point t into a blend unit j are determined by start and finish time of blend run-mode at corresponding event point n. Eqs. (5e)-(5f) force start and finish time to be zero at event-point t if the event-point t is not assigned to any event-point n. Similarly, he complicating variables associated with blend component tanks are determined by Eqs. (6a)-(7f) as shown below. Aggregated component s stream connecting PN and BN sub-networks is split into multiple streams connecting suitable component tanks as expressed by Eqs. (6a)-(6b).

$$KCf_{s,t}^{B} \le \sum_{k \in K_s, j \in J_{dmy}} Kif_{s,j,k,n} + M\left(1 - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{6a}$$

$$KCf_{s,t}^{B} \ge \sum_{k \in K_s, j \in J_{dmy}} Kif_{s,j,k,n} - M\left(1 - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{6b}$$

$$KCf_{s,t}^{B} \le \sum_{k \in K_s} V_k^{\max} \sum_{n \ge t} xk_{s,n,t}^{B}, \quad \forall s \in S_{bc} \cap S_k, t \in T \tag{6c}$$

$$KCTs_{s,t}^{B} \le Tss_{j,k,n} + UH\left(2 - in_{s,j,k,n} - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N, t \in T, n \ge t \tag{7a}$$

$$KCTs_{s,t}^{B} \ge Tss_{j,k,n} - UH\left(2 - in_{s,j,k,n} - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N, t \in T, n \ge t \tag{7b}$$

$$KCTf_{s,t}^{B} \le Tsf_{j,k,n} + UH\left(2 - in_{s,j,k,n} - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N, t \in T, n \ge t \tag{7c}$$

$$KCTf_{s,t}^{B} \ge Tsf_{j,k,n} - UH\left(2 - in_{s,j,k,n} - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N, t \in T, n \ge t \tag{7d}$$

$$KCTs_{s,t}^{B} \le UH \sum_{n \ge t} xk_{s,n,t}^{B}, \quad \forall s \in S_{bc} \cap S_k, t \in T \tag{7e}$$

$$KCTf_{s,t}^{B} \le UH \sum_{n \ge t} xk_{s,n,t}^{B}, \quad \forall s \in S_{bc} \cap S_k, t \in T \tag{7f}$$

If no transfer of flow exists between BN and PN for component s that can be stored in tanks at transfer event-point t, then the corresponding transfer variables are fixed to zero as enforced by Eqs. (6c), (7e) and (7f).

7.3.2. Relaxed Production Unit Scheduling Problem

In production unit operations, more than one production unit can produce a blend component s and these blend component can be either directly supplied to a blend unit $j' \in J^{BU}$ or can be sent to a blend component tanks but not both. Original complicating variables ($JJf_{s,j,j',n}$, $Kif_{s,j,k,n}$, $in_{s,j',k,n}$, $Tss_{j',k,n}$, $Tsf_{j',k,n}$) are eliminated in model (LPSP) and blend components are defined as products instead of intermediates in PN (Figure 7.3). Depending on if the blend component can be stored in tank or not in BN, following changes are made:

- For blend component s supplied directly to a blend unit j' without storage options and produced by production unit j at event-point n, the flow is represented by $Uofb_{s,j,j',n}$.
- For blend component s that can be stored into tanks and produced by production unit j at event-point n, the flow is given by $Uof_{s,j,n}$.

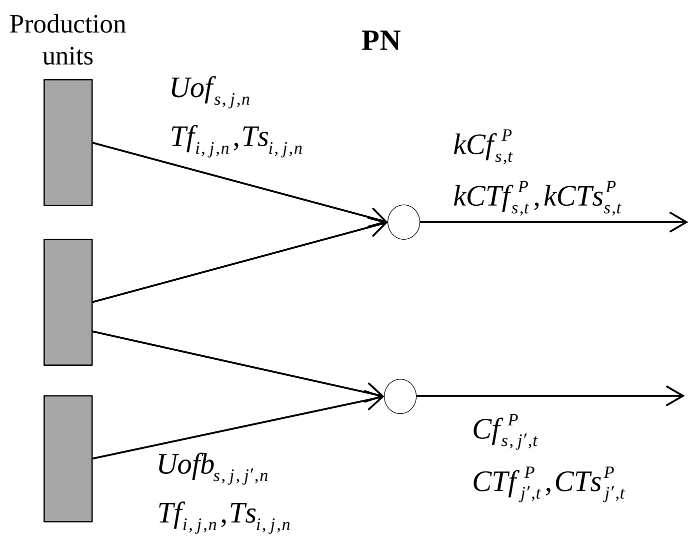

Figure 7.3 Production unit operations network

The material balance constraints (3a) in section 4.3 for blend component s and for production unit j and valid inequality (42) are updated as to define blend components as raw material for PN as follows:

$$\sum_{i \in I_j} bp_{s,i,j,n} = \sum_{k \in K_j^{pk} \cap K_s} Kif_{s,j,k,n} + \sum_{j' \in J_j^{seq} \cap J_s^c} JJf_{s,j,j',n} + Uof_{s,j,n} + \sum_{j' \in J^{BU} \cap J_s^c} Uofb_{s,j,j',n}, \quad \forall s \in S, j \in J_s^p, n \in N \qquad (3a')$$

$$\sum_{k \in K_j^{kp} \cap K_s} Kof_{s,k,j,n} + \sum_{j' \in J_j^{seq} \cap J_s^p} JJf_{s,j',j,n} + Uif_{s,j,n} \leq \sum_{k \in K_j^{pk} \cap K_s} Kif_{s,j,k,n} + \sum_{j' \in J_j^{seq} \cap J_s^c} JJf_{s,j,j',n} + Uof_{s,j,n} + \sum_{j' \in J^{BU} \cap J_s^c} Uofb_{s,j,j',n}, \quad \forall j \in J, n \in N \qquad (42')$$

To determine if transfer happening between production unit j and blend unit j' at event-point n, a binary variable $a_{j,j',n}$ is introduced. This variable is needed because refinery might have multiple blend units that can consume blend component s. According to Eq. (8), if production unit j is supplying a component to a blend unit j' at event-point n then binary variable $a_{j,j',n}$ is equal to 1.

$$Uofb_{s,j,j',n} \le \left(\rho b_{s,j'}^{\max} RB_{j'}^{\max} UH\right) a_{j,j',n}, \quad \forall s \in S_c / S_k, j \in J_s^p, j' \in J^{bu} \cap J_s^c, n \in N \tag{8}$$

Here, $\rho b_{s,j}^{\max}$ is maximum blend recipe component s and $RB_j^{\max}$ is maximum blending rate for blend unit j and they are determined using the available information about blending operations. Eqs. (9a)-(9d) determines maximum/minimum blend recipe for components and maximum/minimum production rate for each blend units, respectively.

$$\rho b_{s,j}^{\max} = \max_{i \in I_j}\left(\rho_{s,i}^{c,\max}\right), \quad \forall s \in S_{bc}, j \in J^{BU} \cap J_s^c \tag{9a}$$

$$\rho b_{s,j}^{\min} = \min_{i \in I_j}\left(\rho_{s,i}^{c,\min}\right), \quad \forall s \in S_{bc}, j \in J^{BU} \cap J_s^c \tag{9b}$$

$$RB_j^{\max} = \max_{i \in I_j}\left(R_{i,j}^{\max}\right), \quad \forall j \in J^{BU} \tag{9c}$$

$$RB_j^{\min} = \min_{i \in I_j}\left(R_{i,j}^{\min}\right), \quad \forall j \in J^{BU} \tag{9d}$$

The flow of a component into sub-network BN is bounded by total storage capacity in BN at any given time as stated by Eq. (10).

$$Uof_{s,j,n} \le \sum_{k \in K_s} V_k^{\max} \sum_{i \in I_j \cap I_s^p} wv_{i,j,n}, \quad \forall s \in S_{bc} \cap S_k, j \in J_s^p, n \in N \tag{10}$$

To obtain the complicating variables in terms of transfer event-point t, constraints similar to (2a)-(2g) are added to production scheduling problem. For a blend unit that receives at least one blend component directly from PN, Eqs. (11a)-(11g) are used and for a blend component that has storage options available in BN, Eqs. (12a)-(12g) are used to determine active transfer flow at event-point t. Here, variables $x_{j,n,t}$ and $xk_{s,n,t}$ are continuous 0-1 variables.

$$x_{j,n,t}^P \le \sum_{j' \in J_j^{seq}} a_{j',j,n}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{11a}$$

$$x_{j,n,t}^{P} \le 1 - \sum_{t'<t,t'\le n} x_{j,n,t'}^{P}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{11b}$$

$$x_{j,n,t}^{P} \le 1 - \sum_{n'<n,n'\ge t} x_{j,n',t}^{P}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \ge t \tag{11c}$$

$$x_{j,n,t}^{P} \ge a_{j',j,n} - \sum_{t'<t,t'\le n} x_{j,n,t'}^{P} - \sum_{n'<n,n'\ge t} x_{j,n,t'}^{P}, \quad \forall j \in J^{BU} \cap J^{ds}, j' \in J_j^{seq}, n \in N, t \in T, n \ge t \tag{11d}$$

$$\sum_{n\ge t'} x_{j,n,t'}^{P} \le \sum_{n\ge t} x_{j,n,t}^{P}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T, t' \in T, t' > t \tag{11e}$$

$$x_{j,n,t}^{P} + x_{j,n',t'}^{P} \le 1, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, n' \in N, n < n', t \in T, t' \in T, t' < t, n \ge t, n' \ge t, n' \ge t', n \ge t' \tag{11f}$$

$$x_{j,n,t}^{P} = 0, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n < t \tag{11g}$$

$$xk_{s,n,t}^{P} \le \sum_{i\in I_s^p, j\in J_i} wv_{i,j,n}, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{12a}$$

$$xk_{s,n,t}^{P} \le 1 - \sum_{t'<t,t'\le n} xk_{s,n,t'}^{P}, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{12b}$$

$$xk_{s,n,t}^{P} \le 1 - \sum_{n'<n,n'\ge t} xk_{s,n',t}^{P}, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \ge t \tag{12c}$$

$$xk_{s,n,t}^{P} \ge \sum_{i\in I_s^p} wv_{i,j,n} - \sum_{t'<t,t'\le n} xk_{s,n,t'}^{P} - \sum_{n'<n,n'\ge t} xk_{s,n,t'}^{P}, \quad \forall s \in S_{bc} \cap S_k, j \in J_s^p, n \in N, t \in T, n \ge t \tag{12d}$$

$$\sum_{n\ge t'} xk_{s,n,t'}^{P} \le \sum_{n\ge t} xk_{s,n,t}^{P}, \quad \forall s \in S_{bc} \cap S_k, t \in T, t' \in T, t' > t \tag{12e}$$

$$xk_{s,n,t}^{P} + xk_{s,n',t'}^{P} \le 1, \quad \forall s \in S_{bc} \cap S_k, n \in N, n' \in N, n < n', t \in T, t' \in T, t' < t, n \ge t, n' \ge t, n' \ge t', n \ge t' \tag{12f}$$

$$xk_{s,n,t}^{P} = 0, \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n < t \tag{12g}$$

Similar to Eqs. (4a)-(7f) used to determine flow amounts and duration of components flow into BN, Eqs. (13a)-(16f) are used to determine the complicating variables associated with transfer event as follows:

$$Cf_{s,j,t}^{P} \le \sum_{j'\in J_s^p} Uofb_{s,j',j,n} + \rho b_{s,j}^{\max} RB_j^{\max} UH\left(1 - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, n \in N, t \in T, n \ge t \tag{13a}$$

$$Cf_{s,j,t}^{P} \ge \sum_{j'\in J_s^p} Uofb_{s,j',j,n} - \rho b_{s,j}^{\max} RB_j^{\max} UH\left(1 - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, n \in N, t \in T, n \ge t \tag{13b}$$

$$Cf_{s,j,t}^{P} \le \rho b_{s,j}^{\max} RB_j^{\max} UH \sum_{n\ge t} x_{j,n,t}^{P}, \quad \forall j \in J^{BU}, s \in S_j^c / S_k, t \in T \tag{13c}$$

$$CTs_{j,t}^{P} \le Ts_{i,j',n} + UH\left(3 - wv_{i,j',n} - a_{j',j,n} - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU} \cap J^{ds}, j' \in J_{j}^{seq}, i \in I_{j'}, n \in N, t \in T, n \ge t \tag{14a}$$

$$CTs_{j,t}^{P} \ge Ts_{i,j',n} - UH\left(3 - wv_{i,j',n} - a_{j',j,n} - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU} \cap J^{ds}, j' \in J_{j}^{seq}, i \in I_{j'}, n \in N, t \in T, n \ge t \tag{14b}$$

$$CTf_{j,t}^{P} \le Tf_{i,j',n} + UH\left(3 - wv_{i,j',n} - a_{j',j,n} - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU} \cap J^{ds}, j' \in J_{j}^{seq}, i \in I_{j'}, n \in N, t \in T, n \ge t \tag{14c}$$

$$CTf_{j,t}^{P} \ge Tf_{i,j',n} - UH\left(3 - wv_{i,j',n} - a_{j',j,n} - x_{j,n,t}^{P}\right), \quad \forall j \in J^{BU} \cap J^{ds}, j' \in J_{j}^{seq}, i \in I_{j'}, n \in N, t \in T, n \ge t \tag{14d}$$

$$CTf_{j,t}^{P} \le UH \sum_{n \ge t} x_{j,n,t}^{P}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T \tag{14e}$$

$$CTf_{j,t}^{P} \le UH \sum_{n \ge t} x_{j,n,t}^{P}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T \tag{14f}$$

$$KCf_{s,t}^{P} \le \sum_{j \in J_{s}^{P}} Uof_{s,j,n} + \sum_{k \in K_{s}} V_{k}^{\max} \cdot \left(1 - xk_{s,n,t}^{P}\right), \quad \forall s \in S_{bc} \cap S_{k}, n \in N, t \in T, n \ge t \tag{15a}$$

$$KCf_{s,t}^{P} \ge \sum_{j \in J_{s}^{P}} Uof_{s,j,n} - \sum_{k \in K_{s}} V_{k}^{\max} \cdot \left(1 - xk_{s,n,t}^{P}\right), \quad \forall s \in S_{bc} \cap S_{k}, n \in N, t \in T, n \ge t \tag{15b}$$

$$KCf_{s,t}^{P} \le \sum_{k \in K_{s}} V_{k}^{\max} \sum_{n \ge t} xk_{s,n,t}^{P}, \quad \forall s \in S_{bc} \cap S_{k}, t \in T \tag{15c}$$

$$KCTs_{s,t}^{P} \le Ts_{i,j,n} + UH\left(2 - wv_{i,j,n} - xk_{s,n,t}^{P}\right), \quad \forall s \in S_{bc}, \cap S_{k}, i \in I_{s}^{p}, j \in J_{i}, n \in N, t \in T, n \ge t \tag{16a}$$

$$KCTs_{s,t}^{P} \ge Ts_{i,j,n} - UH\left(2 - wv_{i,j,n} - xk_{s,n,t}^{P}\right), \quad \forall s \in S_{bc}, \cap S_{k}, i \in I_{s}^{p}, j \in J_{i}, n \in N, t \in T, n \ge t \tag{16b}$$

$$KCTf_{s,t}^{P} \le Tf_{i,j,n} + UH\left(2 - wv_{i,j,n} - xk_{s,n,t}^{P}\right), \quad \forall s \in S_{bc}, \cap S_{k}, i \in I_{s}^{p}, j \in J_{i}, n \in N, t \in T, n \ge t \tag{16c}$$

$$KCTf_{s,t}^{P} \ge Tf_{i,j,n} - UH\left(2 - wv_{i,j,n} - xk_{s,n,t}^{P}\right), \quad \forall s \in S_{bc}, \cap S_{k}, i \in I_{s}^{p}, j \in J_{i}, n \in N, t \in T, n \ge t \tag{16d}$$

$$KCTs_{s,t}^{P} \le UH \sum_{n \ge t} xk_{s,n,t}^{P}, \quad \forall s \in S_{bc} \cap S_{k}, t \in T \tag{16e}$$

$$KCTf_{s,t}^{P} \le UH \sum_{n \ge t} xk_{s,n,t}^{P}, \quad \forall s \in S_{bc} \cap S_{k}, t \in T \tag{16f}$$

As mention before, the multiple streams for a blend component are aggregated into one stream to reduce complicating variables in augmented Lagrangian decomposition algorithm. Thus, the flow of component s between PN and BN at transfer event-point t is determined in Eqs. (13a), (13b), (15a), and (15b) by adding component s flow from multiple production units. More ever, flow of component s to a blend-unit j is bounded by maximum consumption rate ($\rho b_{s,j}^{\max} RB_{j}^{\max} UH$)

and flow to storage tanks is limited by maximum available storage capacity ($\sum_{k\in K_s} V_s^{\max}$). Duration of a transfer event between PN and a blend unit j is determined by the start and finish time of run-modes of production units that are supplying material to the blend unit (i.e. $a_{j',j,n}$ = 1). In addition to determine the timing variables for a direct flow of components between a production unit and a blend unit at transfer event-point t , Eqs. (14a)-(14d) enforces that same start and finish time for flow of all components that can be consumed by the blend unit. Similarly, the (16a)-(16d) assigns start and finish time for flow of components that can be stored in tanks. If no flow transfer between PN and BN at event-point t , then corresponding linking variables are zero as enforced by Eqs. (13c),(14e),(14f), (15c), (16e), and (16f).

7.4. Strengthening the Augmented Lagrangian Decomposition

When refinery operations network is decomposed, information regarding the operational and logistics rules that govern flow of components between PN and BN is lost. To incorporate many of these rules without destroying separability, additional constraints are included in (LBSP) and (LPSP) and presented in section 7.4.2. These constraints speed up convergence to a feasible solution in augmented Lagrangian optimization algorithm and in most instances push feasible solution closer to a global optimal solution. To further improve performance of augmented Lagrangian optimization, valid inequalities can be added to improve computational efficiency of decomposed models and they are presented in Appendix Chapter 7 for model (LBSP) and (LPSP).

To improve integration between PN and BN, production capacity limitation of production unit operations should be obtained before start of the optimization algorithm.

7.4.1. Preprocessing Step

When refiner network is decomposed into (BN) and (PN), the resulting decomposed scheduling problem (LBSP) does not take into consideration the production capacity limitation of production unit operations. Therefore, a pre-processing step before the start of the algorithm is needed to determine maximum and minimum production rate of each blend component.

Relaxed production unit operation scheduling model (LPSP) is optimized for one of the following objective function to obtain maximum and minimum production rate for a blend component s.

$$z = Cf_{s,j,T} - \left(CTf_{j,T} - CTs_{j,T}\right), \quad \forall s \in S_{bc} \setminus S_k, j \in J_s^c \tag{17a}$$

$$z = KCf_{s,T} - \left(KCTf_{s,T} - KCTs_{s,T}\right), \quad \forall s \in S_{bc} \cap S_k \tag{17b}$$

Objective function (17a) is maximized to determine the maximum production flow rate of a component s to a blend unit j and Eq. (17b) is used to obtain the maximum production rate for components that have storage options available. The minimum flow rate would always be zero unless constraint (18a)-(18b) is added to the model to require flow to happen.

$$\sum_n x_{j,n,T} \geq 1, \quad \forall s \in S_{bc} \setminus S_k, j \in J_s^c \tag{18a}$$

$$\sum_n xk_{s,n,T} \geq 1, \quad \forall s \in S_{bc} \cap S_k \tag{18b}$$

Eq. (18a) requires the flow of component s from PN to a blend unit at transfer event-point T and for component that can be stored in tanks, Eq. (18b) enforces flow to happen at event-point T. For blend units that receive all the components directly from production unit operations without any storage, the maximum and minimum total flow rate into a blend unit is determined using objective function given in Eq. (19).

$$z = \sum_{s \in S_{bc}} \left\{Cf_{s,j,T} - \left(CTf_{j,T} - CTs_{j,T}\right)\right\}, \quad \forall j \in J^{BU} \cap J^{ads} \tag{19}$$

Upon solving the pre-processing problems, we obtain values of variables { $Cf^P_{s,j,t}$, $CTs^P_{j,t}$, $CTf^P_{j,t}$, $x^P_{j,n,t}$, $KCf^P_{s,t}$, $KCTs^P_{s,t}$, $KCTf^P_{s,t}$, $xk^P_{s,n,t}$ } from the optimal solution.

$$RPu_{s,j} = \frac{Cf^P_{s,j,T} \sum_n x^P_{j,n,T}}{Tf^P_{j,T} - Ts^P_{j,T}}, \quad \forall j \in J^{BU}, s \in S_j^c \setminus S_k \tag{20a}$$

$$RPk_s = \frac{KCf^P_{s,j,T} \sum_n xk^P_{s,n,T}}{KTf^P_{s,T} - KTs^P_{s,T}}, \quad \forall s \in S_{bc} \cap S_k \tag{20b}$$

$$RP_j = \frac{\sum_{s \in S^c_j} Cf^P_{s,j,T} \sum_n x^P_{j,n,T}}{Tf^P_{j,T} - Ts^P_{j,T}}, \quad \forall j \in J^{BU} \cap J^{ads} \tag{20c}$$

Flow rate for a component from PN to a blend unit is calculated using Eq. (20a), total flow rate into the blend unit is determined using Eq. (20c) and Eq. (20b) is used for components that are stored in tanks.

7.4.2. Strengthening Decomposed Problems

Many constraints that are unnecessary or redundant in full-scale model (P) are needed in decomposed model to obtain a solution that is much more align with a feasible solution of full-scale problem and to speed up convergence to a feasible solution in augmented Lagrangian algorithm. In this section, we propose constraints that improve performance of the decomposition algorithm.

7.4.2.1. Relaxed Blend Scheduling Problem

Lower bound on the duration of the component flow from PN to BN is determined by minimum run-length (RL) requirement for production units in PN. Thus, constraints (21b) similar to Eq. (14) of original model in section 4.3 is included in relaxed problem.

$$Tsf_{j,k,n} - Tss_{j,k,n} \geq RL \cdot in_{s,j,k,n} - UH\left(1 - in_{s,j,k,n}\right), \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N \tag{21b}$$

For a case when more than one blend component can be stored in component tanks in BN and if these blend components are produced by production units in PN that are interconnected, that is these production units must operate at the same time, then components produced by these units would be supplied to BN at the same time.

$$in_{s,j,k,n} \leq \sum_{k' \in K_{s'}} in_{s',j,k',n}, \quad \forall s \in S_{bc}, s' \in S_{bc}, k \in K_s, j \in J_{dmy}, n \in N, cont_{s,s'} > 0 \tag{22}$$

$$\sum_{k\in K_s,k'\in K_{s'}} in_{s,j,k,n} \geq \sum_{k'\in K_{s'}} in_{s',j,k',n}, \quad \forall s\in S_{bc}, s'\in S_{bc}, j\in J_{dmy}, n\in N, cont_{s,s'}>0 \tag{23}$$

Here, parameter $cont_{s,s'}$ is equal to 1 if components s and s' are produced by units in PN that are interconnected. Eqs. (22)-(23) states that if a blend unit *s* is being supplied to a component tank at event-point *n* then component s' must also be supplied to at least one suitable component tank at event-point *n*. Timing constraints for these connected blend components are given in Eqs. (24a)-(24d) which enforce same start and finish time for component flow at event-point *n*.

$$\begin{aligned} Tss_{j,k,n} &\leq Tss_{j,k',n} + UH\left(2 - in_{s,j,k,n} - in_{s',j,k',n}\right), \\ &\forall s\in S_{bc}, s'\in S_{bc}, k\in K_s, k'\in K_{s'}, j\in J_{dmy}, n\in N, cont_{s,s'}>0 \end{aligned} \tag{24a}$$

$$\begin{aligned} Tss_{j,k,n} &\geq Tss_{j,k',n} - UH\left(2 - in_{s,j,k,n} - in_{s',j,k',n}\right), \\ &\forall s\in S_{bc}, s'\in S_{bc}, k\in K_s, k'\in K_{s'}, j\in J_{dmy}, n\in N, cont_{s,s'}>0 \end{aligned} \tag{24b}$$

$$\begin{aligned} Tsf_{j,k,n} &\leq Tsf_{j,k',n} + UH\left(2 - in_{s,j,k,n} - in_{s',j,k',n}\right), \\ &\forall s\in S_{bc}, s'\in S_{bc}, k\in K_s, k'\in K_{s'}, j\in J_{dmy}, n\in N, cont_{s,s'}>0 \end{aligned} \tag{24c}$$

$$\begin{aligned} Tsf_{j,k,n} &\geq Tsf_{j,k',n} - UH\left(2 - in_{s,j,k,n} - in_{s',j,k',n}\right), \\ &\forall s\in S_{bc}, s'\in S_{bc}, k\in K_s, k'\in K_{s'}, j\in J_{dmy}, n\in N, cont_{s,s'}>0 \end{aligned} \tag{24d}$$

Furthermore, if multiple tanks are available to store a blend component, then following sequence constraint (25) are needed as it captures what happens in during production unit operations in PN.

$$Tss_{j,k,n+1} \geq Tsf_{j,k',n}, \quad \forall s\in S_{bc}, k\in K_s, k'\in K_s, k\neq k', j\in J_{dmy}, n<N \tag{25}$$

Set J^{ds} includes all blend units that receive at least one blend component directly from production unit operations without any storage. If all the components consumed by blend unit j are supplied directly without storage, then that blend unit belongs to set J^{ads}.

The production capacity of blend components determined in pre-processing step is incorporated into relaxed blend scheduling by removing the big-M term from model.

$$Kif_{s,j,k,n} \leq RPk_s^{\max}\left(Tsf_{j,k,n} - Tss_{j,k,n}\right) + RPk_s^{\max} UH\left(1 - in_{s,j,k,n}\right), \quad \forall s\in S_{bc}, k\in K_s, j\in J_{dmy}, n\in N \tag{1'}$$

$$Cf_{s,j,t}^{B} \leq Uif_{s,j,n} + RPu_{s,j}^{\max} UH\left(1 - x_{j,n,t}^{B}\right), \quad \forall j\in J^{BU}, s\in S_j^c / S_k, n\in N, t\in T, n\geq t \tag{4a'}$$

$$Cf_{s,j,t}^{B} \geq Uif_{s,j,n} - RPu_{s,j}^{\max} UH\left(1 - x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, n \in N, t \in T, n \geq t \tag{4b'}$$

$$Cf_{s,j,t}^{B} \leq RPu_{s,j}^{\max} UH \sum_{n \geq t} x_{j,n,t}^{B}, \quad \forall j \in J^{BU}, s \in S_j^c / S_k, t \in T \tag{4c'}$$

$$KCf_{s,t}^{B} \leq \sum_{k \in K_s, j \in J_{dmy}} Kif_{s,j,k,n} + RPk_s^{\max} UH\left(1 - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \geq t \tag{6a'}$$

$$KCf_{s,t}^{B} \geq \sum_{k \in K_s, j \in J_{dmy}} Kif_{s,j,k,n} - RPk_s^{\max} UH\left(1 - xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc} \cap S_k, n \in N, t \in T, n \geq t \tag{6b'}$$

Eqs. (1), (4a), (4b), (4c), (6a), and (6b) are replaced with updated constraints (1'), (4a'), (4b'), (4c'), (6a'), and (6b'). Upper and lower limit on flow into BN from PN at transfer event-point t is given by Eqs. (26a)-(28b).

$$Cf_{s,j,t}^{B} \leq RPu_{s,j}^{\max}\left(CTf_{j,t}^{B} - CTs_{j,t}^{B}\right) + RPu_{s,j}^{\max} UH\left(1 - \sum_{n \geq t} x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, t \in T \tag{26a}$$

$$Cf_{s,j,t}^{B} \geq RPu_{s,j}^{\min}\left(CTf_{j,t}^{B} - CTs_{j,t}^{B}\right) - RPu_{s,j}^{\max} UH\left(1 - \sum_{n \geq t} x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU}, s \in S_j^c / S_k, t \in T \tag{26b}$$

$$KCf_{s,t}^{B} \leq RPk_s^{\max}\left(KCTf_{s,t}^{B} - KCTs_{s,t}^{B}\right) + RPk_s^{\max} UH\left(1 - \sum_{n \geq t} xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc} \cap S_k, t \in T \tag{27a}$$

$$KCf_{s,t}^{B} \geq RPk_s^{\min}\left(KCTf_{s,t}^{B} - KCTs_{s,t}^{B}\right) - RPk_s^{\max} UH\left(1 - \sum_{n \geq t} xk_{s,n,t}^{B}\right), \quad \forall s \in S_{bc} \cap S_k, t \in T \tag{27b}$$

$$\sum_{s \in S_j^c / S_k} Cf_{s,j,t}^{B} \leq RP_j^{\max}\left(CTf_{j,t}^{B} - CTs_{j,t}^{B}\right) + RP_j^{\max} UH\left(1 - \sum_{n \geq t} x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU} \cap J^{ads}, t \in T \tag{28a}$$

$$\sum_{s \in S_j^c / S_k} Cf_{s,j,t}^{B} \geq RP_j^{\min}\left(CTf_{j,t}^{B} - CTs_{j,t}^{B}\right) - RP_j^{\max} UH\left(1 - \sum_{n \geq t} x_{j,n,t}^{B}\right), \quad \forall j \in J^{BU} \cap J^{ads}, t \in T \tag{28b}$$

Flow of component s to blend unit j is bounded by product of duration of flow and maximum/minimum production rate ($RPu_{s,j}^{\max} / RPu_{s,j}^{\min}$) as limited by Eqs. (26a) and (26b), respectively. If no flow exists between PN and a blend unit (i.e. $\sum_n x_{j,n,t} = 0$) then inequality in (26a)-(26b) is relaxed. For a blend component that can be stored in tanks, if no flow exists between PN and BN for a (i.e. $\sum_n xk_{s,n,t} = 0$) then inequalities (27a)-(27b) are relaxed. Similarly, for blend units that receive all their components directly from production unit operations, the total flow amount is bounded by upper and lower limits as enforced by Eqs. (28a)-(28b).

A lower bound of (LBSP) is obtained, when the model (LBSP) is solved to optimality with all the Lagrangian and augmented penalty parameter fixed to zero. Thus, solution $\left(f^{BSP}\right)^{(m=0)}$ at algorithm count $m=0$ gives the best lower bound of (LBSP) and Eq. (29) is added to model.

$$f^{BSP} \geq \left(f^{BSP}\right)^{(m=0)}, \quad if \quad m \geq 1 \tag{29}$$

For a refinery that has multiple dedicated tanks for blend component storage, Eq. (30) is added to model when algorithm iteration count is $m \geq 1$.

$$\sum_{n<N} in_{s,j,k,n} \geq 1, \quad \forall s \in S_{bc}, k \in K_s^h, j \in J_{dmy}, m \geq 1 \quad if \ \sum_n in_{s,j,k,n}^{(m=0)} > 0 \tag{30}$$

The blend scheduling problem become difficult to solve as the additional flexibility of a parallel blend component tanks is introduced. To reduce this complexity, Eq. (30) is added for parallel blend component tanks.

In our work, since we do not maintain minimum inventory levels for components or blend products, following constraints can be added to simplify the blend scheduling problem.

If one were to require either minimum inventory level for blend components but not finished blend product, then constraint (31) can be added to the model without compromising the optimal solution. Finished product blend tanks cannot have simultaneous loading and unloading operations for security reasons and to allow time for mixing and product specification certificate analysis. For this reason, if a blend unit is receiving all of its blend components from storage tanks, then this blend unit would not be active during last event point N because the product produce will not contribute to demand orders fulfillment and there is no minimum inventory requirement for products. We add Eqs. (31) to (LBSP) to fix blend unit allocation $wv_{i,j,N}$ to zero.

$$wv_{i,j,n} = 0, \quad \forall j \in J^{ads}, i \in I_j, n = N \tag{31}$$

Furthermore, linking variables are compared for transfer event-point t and not event-point n, the scheduling problem corresponding to sub-network PN can be solved using less number of event-points n ($ep_{PN}^{\min}$) than scheduling problem corresponding to sub-network BN. When $ep_{PN}^{\min} < ep_{bn}^{\min}$, then following variables are zero at last event-point in (LBSP) when no minimum inventory level requirement for components and blend products.

$$in_{s,j,k,n} = 0, \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, n = N, if \ ep_{PN}^{\min} < ep_{bn}^{\min} \tag{32a}$$

$$in_{s,j,k,n} = 0, \quad \forall j \in J^{BU} \cap J^{ds}, s \in S_j^p, k \in K_s, n = N, if \ ep_{PN}^{\min} < ep_{bn}^{\min} \tag{32b}$$

$$wv_{i,j,n} = 0, \quad \forall j \in J^{BU} \cap J^{ds}, i \in I_j, n = N, if \ ep_{PN}^{\min} < ep_{bn}^{\min} \tag{32c}$$

$$x_{j,n,t} = 0, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T, n = N, if \ ep_{PN}^{\min} < ep_{bn}^{\min} \tag{32d}$$

$$xk_{s,n,t} = 0, \quad \forall s \in S_{bc}, t \in T, n = N, if \ ep_{PN}^{\min} < ep_{bn}^{\min} \tag{32e}$$

Cardinality of set $|T| = ep_{PN}^{\min}$ and transfer event-point t only corresponds to an event-point $n \geq t$. When $ep_{PN}^{\min} < ep_{bn}^{\min}$, flow of components into the BN happens only to satisfy finished product demand, and Eqs. (32a)-(32e) fixes some binary and 0-1 continuous variables to zero for $n = N$. If one were to require minimum inventory level for products and components, then these constraints should not be included in the model.

7.4.2.2. Relaxed Production unit Scheduling Problem

To insure that demand commitments for finished blend product are met and excess inventory is minimized in BN, Eqs. (33a) –(33b) are included when parameter is r not zero. The parameter r is different than algorithm iteration count m and will be defined in section 7.5.

$$\begin{aligned} &\sum_{s \in S_{bc} \cap S_k, j \in J_s^p, n < ep_{bn}^{\min}} Uof_{s,j,n} + \sum_{s \in S_{bc} / S_k, j \in J_s^p, j' \in J^{BU} \cap J_s^c, n < ep_{bn}^{\min}} Uofb_{s,j,j',n} \\ &\qquad \geq \sum_{s \in S_A \cap S^{bn}, o \in O_s} D_{o,s}^{-} - \sum_{s \in S^{bn}, k \in K_s} sto_{s,k} - DmP_{bn}, \quad \forall bn \in BN, r \geq 1 \end{aligned} \tag{33a}$$

$$\sum_{s\in S_{bc}\cap S_k, j\in J_s^P, n<ep_{bn}^{\min}} Uof_{s,j,n} + \sum_{s\in S_{bc}/S_k, j\in J_s^P, j'\in J^{BU}\cap J_s^c, n<ep_{bn}^{\min}} Uofb_{s,j,j',n} \leq \sum_{s\in S_A\cap S^{bn}, o'\in O_s} D_{o',s}^{+} + InvtP_{bn}, \quad \forall bn\in BN, r\geq 1 \tag{33b}$$

Each independent sub-networks of BN has their own set of finished blend product demand orders and initial storage inventory. Eq. (33a) requires production unit operation to produce at least minimum amount of total components to satisfy demand for each blending operations sub-network. More ever, if the total demand cannot be met, then a positive slack variable DmP_{bn} is added to facilitate feasible solution. Eq. (33b) limits the total production of blend components as it is bounded by maximum demand for finished blend products. For cases where production exceeds, a positive slack variable $InvtP_{bn}$ is introduced and is later penalized in objective function.

To ensure that demand is met on time and to improve performance of model, constraints (34a) and (35c) are added to the model when parameter is $r\geq 1$.

$$CTs_{j,t}^{P} \leq \overline{CTs}_{j,t}^{B} + esP_{j,t}, \quad \forall bn\in BN, j\in J_{bn}^{BU}, t\in T, t<ep_{bn}^{\min}, \sum_{s}\overline{Cf}_{s,j,t}^{B} > 0, r\geq 1 \tag{34a}$$

$$CTs_{j,t}^{P} \geq \overline{CTs}_{j,t}^{B} - esP_{j,t}, \quad \forall bn\in BN, j\in J_{bn}^{BU}, t\in T, t<ep_{bn}^{\min}, \sum_{s}\overline{Cf}_{s,j,t}^{B} > 0, r\geq 1 \tag{34b}$$

$$CTf_{j,t}^{P} \leq \max_{t'<ep_{bn}^{\min}} + leP_{j}, \quad \forall bn\in BN, j\in J_{bn}^{BU}, t\in T, t<ep_{bn}^{\min}, \sum_{s}\overline{Cf}_{s,j,t}^{B} > 0, r\geq 1 \tag{34c}$$

$$KCTs_{s,t}^{P} \leq \overline{KCTs}_{s,t}^{B} + eskP_{s,t}, \quad \forall bn\in BN, s\in S_{bc}, t\in T, t<ep_{bn}^{\min}, \overline{KCf}_{s,t}^{B} > 0, r\geq 1 \tag{35a}$$

$$KCTs_{s,t}^{P} \geq \overline{KCTs}_{s,t}^{B} - eskP_{s,t}, \quad \forall bn\in BN, s\in S_{bc}, t\in T, t<ep_{bn}^{\min}, \overline{KCf}_{s,t}^{B} > 0, r\geq 1 \tag{35b}$$

$$KCTf_{s,t}^{P} \leq \max_{t'<ep_{bn}^{\min}}\left(\overline{KCTf}_{s,t}^{B}\right) + lekP_{s}, \quad \forall bn\in BN, s\in S_{bc}, t\in T, t<ep_{bn}^{\min}, \overline{KCf}_{s,t}^{B} > 0, r\geq 1 \tag{35c}$$

Constraints (34a)-(34b) and (35a)-(35b) encourage the start and finish time of flow to be near the corresponding timing variables' solution obtain solving (LBSP). Here $esP_{j,t}$, $eskP_{s,t}$, leP_{j}, and $lekP_{s}$ are positive slack variables which are penalized in objective function.

Constraint (36) requires the objective function to be bounded by $\left(f^{PSP}\right)^{r=0}$, similar to Eq. (73) in (LBSP).

$$f^{PSP} \geq \left(f^{PSP}\right)^{(r=0)}, \quad if \quad r \geq 1 \tag{36}$$

$$\sum_{n \in N, n \geq t} x^P_{j,n,t} = 1, \quad \forall j \in J^{BU}, t \in T, \sum_s \overline{Cf}^B_{s,j,t} > 0, r \geq 1 \tag{37a}$$

$$\sum_{n \in N, n \geq t} xk^P_{s,n,t} = 1, \quad \forall s \in S_{bc} \cap S_k, t \in T, \overline{KCf}^B_{s,t} > 0, r \geq 1 \tag{37b}$$

Since PN is producing blend components for BN, we add following requirement (37a)-(37b) in model (LBSP) when flow variables are nonzero in BN for transfer event-point t.

The slack variables ($DmP_{bn}, Invt_{bn}, esP_{j,t}, leP_j, eskP_{s,t}, lekP_s$) are penalized and the updated objective function for the model (LPSP) is given below:

$$\begin{aligned}
\min\ ff^{PSP} = z &- \sum_{s \in S_{bc}, j \in J^{BU}, t} \left(\lambda_{s,j,t} Cf^P_{s,j,t}\right) - \sum_{j \in J^{BU}, t} \left(\pi_{j,t} CTs^P_{j,t} + \tau_{j,t} CTf^P_{j,t}\right) \\
&- \sum_{s \in S_{bc}, t} \left(\mu_{s,t} KCf^P_{s,t} + \nu_{s,t} KCTs^P_{s,t} + \upsilon_{s,t} KCTf^P_{s,t}\right) \\
&+ \sigma \sum_{s \in S_{bc}, j \in J^{BU}, t} \left\{\left(\overline{Cf}^B_{s,j,t} - Cf^P_{s,j,t}\right)^2 - \left(\overline{Cf}^B_{s,j,t} - \overline{Cf}^P_{s,j,t}\right)^2\right\} \\
&+ \sigma \sum_{j \in J^{BU}, t} \left\{\begin{aligned} &\left(\overline{CTs}^B_{j,t} - CTs^P_{j,t}\right)^2 - \left(\overline{CTs}^B_{j,t} - \overline{CTs}^P_{j,t}\right)^2 \\ &+ \left(\overline{CTf}^B_{j,t} - CTf^P_{j,t}\right)^2 - \left(\overline{CTf}^B_{j,t} - \overline{CTf}^P_{j,t}\right)^2 \end{aligned}\right\} \\
&+ \sigma \sum_{s \in S_{bc}, t} \left\{\begin{aligned} &\left(\overline{KCf}^B_{s,t} - KCf^P_{s,t}\right)^2 - \left(\overline{Kif}^B_{s,j',k,n} - \overline{Kif}^P_{s,j',k,n}\right)^2 \\ &+ \left(\overline{KCTs}^B_{s,t} - KCTs^P_{s,t}\right)^2 - \left(\overline{KCTs}^B_{s,t} - \overline{KCTs}^P_{s,t}\right)^2 \\ &+ \left(\overline{KCTf}^B_{s,t} - KCTf^P_{s,t}\right)^2 - \left(\overline{KCTf}^B_{s,t} - \overline{KCTf}^P_{s,t}\right)^2 \end{aligned}\right\} \\
&+ c^3 \sum_{bn \in BN, r \geq 1} \left\{Invt_{bn} + \sum_{s \in S_{bc} \cap S_k, j \in J^P_s, n \geq ep^{\min}_{bn}} Uof_{s,j,n} + \sum_{s \in S_{bc}/S_k, j \in J^P_s, j' \in J^{BU} \cap J^c_s, n \geq ep^{\min}_{bn}} Uofb_{s,j,j',n}\right\} \\
&+ c^{11} \sum_{bn \in BN, r \geq 1} DmP_{bn} + c^{16} \sum_{j \in J^{BU}, t, r \geq 1} esP_{j,t} + c^{17} \sum_{s \in S_{bc} \cap S_k, t, r \geq 1} eskP_{s,t} + c^{18} \sum_{j \in J^{BU}, r \geq 1} leP_j \\
&+ c^{19} \sum_{s \in S_{bc} \cap S_k, r \geq 1} lekP_s + \sum_{j \in J^h, t < ep^{\min}_{bn}, \sum_{j' \in J^h, s} \overline{Cf}^B_{s,j',t} > 0, \sum_s \overline{Cf}^B_{s,j,t} = 0, r \geq 1} c^{20} x_{j,n,t} \\
&+ \sum_{j \in J^h, t < ep^{\min}_{bn}, \sum_{j' \in J^h, s} \overline{Cf}^B_{s,j',t} = 0, r \geq 1} c^1_j x_{j,n,t} + \sum_{j \in J^h, t \geq ep^{\min}_{bn}, r \geq 1} c^1_j x_{j,n,t}
\end{aligned}$$

Objective function ff^{PSP} includes original objective function, augmented Lagrangian terms and additional terms that tries to take into consideration the cost that will be incurred in blending operations by minimizing excess inventory for blend components, under production of blend components and component flow start and finish time violations based on solution of LBSP. This transfer time violation is not penalized heavily as it would be the goal of augmented Lagrangian optimization algorithm to eliminate this violation in order to obtain feasible solution of global problem. The last three terms in the objective function is for parallel blend units. If the material flow is happening to at least one parallel blend unit in BN ($\sum_{s}\overline{Cf}^{B}_{s,j,t} > 0$) at event-point t then the rest of the parallel units that have no flow happening ($\sum_{s}\overline{Cf}^{B}_{s,j,t} = 0$) at that event-point are penalized less heavily than if no flow was happening at any parallel blend unit ($\sum_{s,j}\overline{Cf}^{B}_{s,j,t} = 0$) at event-point t. Cost coefficients $c^{16}, c^{17}, c^{18}, c^{19}$, and c^{20} need to be determined along with Lagrangian multipliers and augmented penalty parameters. The coefficients c^{20} should be $0.50 < c^{20} < c^{1}$ and parameters $c^{16}, c^{17}, c^{18}, c^{19}$ should be within [0,1].

7.5. Augmented Lagrangian Optimization Algorithm

Before proposed augmented Lagrangian optimization algorithm is presented, models under for the algorithm are as follows:

$$
\begin{aligned}
\min\ f^{BSP} = z &+ \sum_{s\in S_{bc},\, j\in J^{BU},\, t} \lambda_{s,j,t} Cf^{B}_{s,j,t} + \sum_{j\in J^{BU},\, t} \left(\pi_{j,t} CTs^{B}_{j,t} + \tau_{j,t} CTf^{B}_{j,t} \right) \\
&+ \sum_{s\in S_{bc},\, t} \left(\mu_{s,t} KCf^{B}_{s,t} + \nu_{s,t} KCTs^{B}_{s,t} + \upsilon_{s,t} KCTf^{B}_{s,t} \right) \\
&+ \sum_{s\in S_{bc},\, j\in J^{BU},\, t} \sigma_{Cf} \left(Cf^{B}_{s,j,t} - \overline{Cf}^{P}_{s,j,t} \right)^2 + \sum_{j\in J^{BU},\, t} \left\{ \sigma_{CTs} \left(CTs^{B}_{j,t} - \overline{CTs}^{P}_{j,t} \right)^2 + \sigma_{CTf} \left(CTf^{B}_{j,t} - \overline{CTf}^{P}_{j,t} \right)^2 \right\} \\
&+ \sum_{s\in S_{bc},\, t} \left\{ \sigma_{KCf} \left(KCf^{B}_{s,t} - \overline{KCf}^{P}_{s,t} \right)^2 + \sigma_{KCTs} \left(KCTs^{B}_{s,t} - \overline{KCTs}^{P}_{s,t} \right)^2 + \sigma_{KCTf} \left(KCTf^{B}_{s,t} - \overline{KCTf}^{P}_{s,t} \right)^2 \right\}
\end{aligned}
$$

s.t. constraints corresponding to units and connections in BN

(LBSP)

and valid inequalities in Appendix Chapter 7

$$
\begin{aligned}
\min\ ff^{PSP} = {} & z - \sum_{s\in S_{bc},\, j\in J^{BU},t} \left(\lambda_{s,j,t} Cf^{P}_{s,j,t}\right) - \sum_{j\in J^{BU},t} \left(\pi_{j,t} CTs^{P}_{j,t} + \tau_{j,t} CTf^{P}_{j,t}\right) \\
& - \sum_{s\in S_{bc},t} \left(\mu_{s,t} KCf^{P}_{s,t} + \nu_{s,t} KCTs^{P}_{s,t} + \upsilon_{s,t} KCTf^{P}_{s,t}\right) \\
& + \sum_{s\in S_{bc},\, j\in J^{BU},t} \sigma_{Cf} \left\{ \left(\overline{Cf}^{B}_{s,j,t} - Cf^{P}_{s,j,t}\right)^2 - \left(\overline{Cf}^{B}_{s,j,t} - \overline{Cf}^{P}_{s,j,t}\right)^2 \right\} \\
& + \sigma \sum_{j\in J^{BU},t} \left\{ \begin{aligned} & \sigma_{CTs} \left\{ \left(\overline{CTs}^{B}_{j,t} - CTs^{P}_{j,t}\right)^2 - \left(\overline{CTs}^{B}_{j,t} - \overline{CTs}^{P}_{j,t}\right)^2 \right\} \\ & + \sigma_{CTf} \left\{ \left(\overline{CTf}^{B}_{j,t} - CTf^{P}_{j,t}\right)^2 - \left(\overline{CTf}^{B}_{j,t} - \overline{CTf}^{P}_{j,t}\right)^2 \right\} \end{aligned} \right\} \\
& + \sigma \sum_{s\in S_{bc},t} \left\{ \begin{aligned} & \sigma_{KCf} \left\{ \left(\overline{KCf}^{B}_{s,t} - KCf^{P}_{s,t}\right)^2 - \left(\overline{Kif}^{B}_{s,j',k,n} - \overline{Kif}^{P}_{s,j',k,n}\right)^2 \right\} \\ & + \sigma_{KCTs} \left\{ \left(\overline{KCTs}^{B}_{s,t} - KCTs^{P}_{s,t}\right)^2 - \left(\overline{KCTs}^{B}_{s,t} - \overline{KCTs}^{P}_{s,t}\right)^2 \right\} \\ & + \sigma_{KCTf} \left\{ \left(\overline{KCTf}^{B}_{s,t} - KCTf^{P}_{s,t}\right)^2 - \left(\overline{KCTf}^{B}_{s,t} - \overline{KCTf}^{P}_{s,t}\right)^2 \right\} \end{aligned} \right\} \\
& + c^3 \sum_{bn\in BN, r\ge 1} \left\{ Invt_{bn} + \sum_{s\in S_{bc}\cap S_k,\, j\in J^{p}_{s},\, n\ge ep^{\min}_{bn}} Uof_{s,j,n} + \sum_{s\in S_{bc}/S_k,\, j\in J^{p}_{s},\, j'\in J^{BU}\cap J^{c}_{s},\, n\ge ep^{\min}_{bn}} Uofb_{s,j,j',n} \right\} \\
& + c^{11} \sum_{bn\in BN, r\ge 1} DmP_{bn} + c^{16} \sum_{j\in J^{BU},t,r\ge 1} esP_{j,t} + c^{17} \sum_{s\in S_{bc}\cap S_k,t,r\ge 1} eskP_{s,t} + c^{18} \sum_{j\in J^{BU},r\ge 1} leP_j + c^{19} \sum_{s\in S_{bc}\cap S_k,r\ge 1} lekP_s \\
& + \sum_{j\in J^h,\, t<ep^{\min}_{bn},\, \sum_{j'\in J^h,s} \overline{Cf}^{B}_{s,j',t}>0,\, \sum_{s} \overline{Cf}^{B}_{s,j,t}=0,\, r\ge 1} c^{20} x_{j,n,t} + \sum_{j\in J^h,\, t<ep^{\min}_{bn},\, \sum_{j'\in J^h,s} \overline{Cf}^{B}_{s,j',t}=0,\, r\ge 1} c^{1}_{j} x_{j,n,t} + \sum_{j\in J^h,\, t\ge ep^{\min}_{bn},\, r\ge 1} c^{1}_{j} x_{j,n,t}
\end{aligned}
$$

s.t. constraints corresponding to units and connections in PN

(LPSP)

and valid inequalities in Appendix Chapter 7

The Model (LBSP) includes equations, objective function and variables present in model (LRP) corresponding to units and connection in sub-network BN and constraints presented in section 7.3.1, section 7.4.2.1. If sub-network BN can be decomposed into smaller structures, then the model (LBSP) will be decomposed such that model (LBSP1) would only include constraints regarding units and connections present in sub-structure BN1, and so on. Meanwhile, the model (LPSP) includes constraints and variables present in (LRP) pertaining to sub-network PN, constraints proposed in sections 7.3.2 and 7.4.2.2, and objective function ff^{PSP}. Due to quadratic terms in the objective function, models (LBSP) and (LPSP) are mixed integer quadratic programming (MIQP) problems.

$$\max z_{s,j} = Cf_{s,j,T} - \left(CTf_{j,T} - CTs_{j,T}\right), \quad \forall s \in S_{bc} \setminus S_k, j \in J_s^c$$

s.t. constraints corresponding to units and connections in PN (PCs-1)

$$\min z_{s,j} = Cf_{s,j,T} - \left(CTf_{j,T} - CTs_{j,T}\right), \quad \forall s \in S_{bc} \setminus S_k, j \in J_s^c$$

s.t. constraints and variables corresponding model (LPSP) and equation (18a) (PCs-2)

Models (PCs-1) and (PCs-2) is used to determine maximum and minimum production rate of blend component s to blend unit j, respectively.

$$\max z_s = KCf_{s,T} - \left(KCTf_{s,T} - KCTs_{s,T}\right), \quad \forall s \in S_{bc} \cap S_k$$

s.t. constraints and variables corresponding model (LPSP) (PKs-1)

$$\min z_s = KCf_{s,T} - \left(KCTf_{s,T} - KCTs_{s,T}\right), \quad \forall s \in S_{bc} \cap S_k$$

s.t. constraints and variables corresponding model (LPSP) and equation (18b) (PKs-2)

The upper and lower limit of production rate for blend components that are stored is obtained solving models (PKs-1) and (PKs-2), respectively, and these problems are optimized for each blend component.

$$\max z_j = \sum_{s \in S_{bc}} \left\{Cf_{s,j,T} - \left(CTf_{j,T} - CTs_{j,T}\right)\right\}, \quad \forall j \in J^{BU} \cap J^{ads}$$

s.t. constraints and variables corresponding model (LPSP) (PJ-1)

$$\min z_j = \sum_{s \in S_{bc}} \left\{ Cf_{s,j,T} - \left(CTf_{j,T} - CTs_{j,T} \right) \right\}, \quad \forall j \in J^{BU} \cap J^{ads}$$

s.t. constraints and variables corresponding model (LPSP) and equation (18a) (PJ-2)

Maximum and minimum production rate limits for a blend unit that receives all of its components straight from production units is determined using models (PJ-1) and (PJ-2), respectively. All the pre-processing MILP models are independent and can be solved in parallel.

Proposed Augmented Lagrangian optimization algorithm is shown in Figure 7.4 and the detailed steps and termination criteria are as follows:

Step 1: Solve preprocessing models (PJ-1), (PJ-2), (PCs-1), (PCs-2), (PKs-1), and (PKs-2) in parallel. Calculate $RPu_{s,j}^{\max}$, $RPu_{s,j}^{\min}$, $RPk_{s}^{\max}$, $RPk_{s}^{\min}$, $RP_{j}^{\max}$, and $RP_{j}^{\min}$.

Step 2: Set $m = 0$, $r = 0$. Initialize Lagrangian multipliers $\lambda_{s,j,t}^{(0)} = 0$, $\pi_{s,j,t}^{(0)} = 0$, $\tau_{s,j,t}^{(0)} = 0$, $\mu_{s,t}^{(0)} = 0$, $\nu_{s,t}^{(0)} = 0$, $\upsilon_{s,t}^{(0)} = 0$ and penalty parameters $\sigma_{Cf}^{(0)}$, $\sigma_{CTs}^{(0)}$, $\sigma_{CTf}^{(0)}$, $\sigma_{KCf}^{(0)}$, $\sigma_{KCTs}^{(0)}$, $\sigma_{KCTf}^{(0)}$. Pick algorithm parameters γ , β , $\varepsilon = 1$, and maximum algorithm count M .

Step 3: For $r = 0$, solve model (LBSP) and (LPSP) in parallel and obtain lower bounds $\left(f^{BSP} \right)^{(m=0)}$, $\left(f^{PSP} \right)^{(r=0)}$, and update $r = r + 1$.

Otherwise if $r \geq 1$, solve (LBSP).

Obtain $\overline{Cf}_{s,j,t}^{B}, \overline{CTs}_{j,t}^{B}, \overline{CTf}_{j,t}^{B}, \overline{KCf}_{s,t}^{B}, \overline{KCTs}_{s,t}^{B}, \overline{KCTf}_{s,t}^{B}$.

Step 4: Solve (LPSP) and obtain $\overline{Cf}_{s,j,t}^{P}, \overline{CTs}_{j,t}^{P}, \overline{CTf}_{j,t}^{P}, \overline{KCf}_{s,t}^{P}, \overline{KCTs}_{s,t}^{P}, \overline{KCTf}_{s,t}^{P}$.

Step 5: Check if violation matrix norm ($\|g\|$) meets tolerance ε . Where,

$$g = \begin{bmatrix} Cf_{s,j,t}^{B} - Cf_{s,j,t}^{P} \\ KCf_{s,t}^{B} - KCf_{s,t}^{P} \\ CTs_{j,t}^{B} - CTs_{j,t}^{P} \\ CTf_{j,t}^{B} - CTf_{j,t}^{P} \\ KCTs_{s,t}^{B} - KCTs_{s,t}^{P} \\ KCTf_{s,t}^{B} - KCTf_{s,t}^{P} \end{bmatrix}$$

If $\|g\| > \varepsilon$, set $m = m+1$. Update multipliers using equation $\lambda^{(m+1)} = \lambda^{(m)} + \sigma^{m}\left(x - xx\right)^{m}$.

If $\|g\|^{m} > \gamma\|g\|^{m-1}$, update $\sigma^{m+1} = \beta\sigma^{m}$ and go to step 3.

Step 7: If $\|g\| \le \varepsilon$, a feasible solution is found and the objective function for refinery operation is calculated as follows:

$$f = f^{PSP} + \sum_{bn \in BN}\left(f_{bn}^{BSP} - \bar{H}\right)$$

Minimization of time horizon is included in (LPSP) and (LBSP) objective function thus makespan H is subtracted from solution of (LBSP).

Step 8: if $m > M$, terminate the algorithm as the maximum iterations limit is reached. Otherwise, go to step 3.

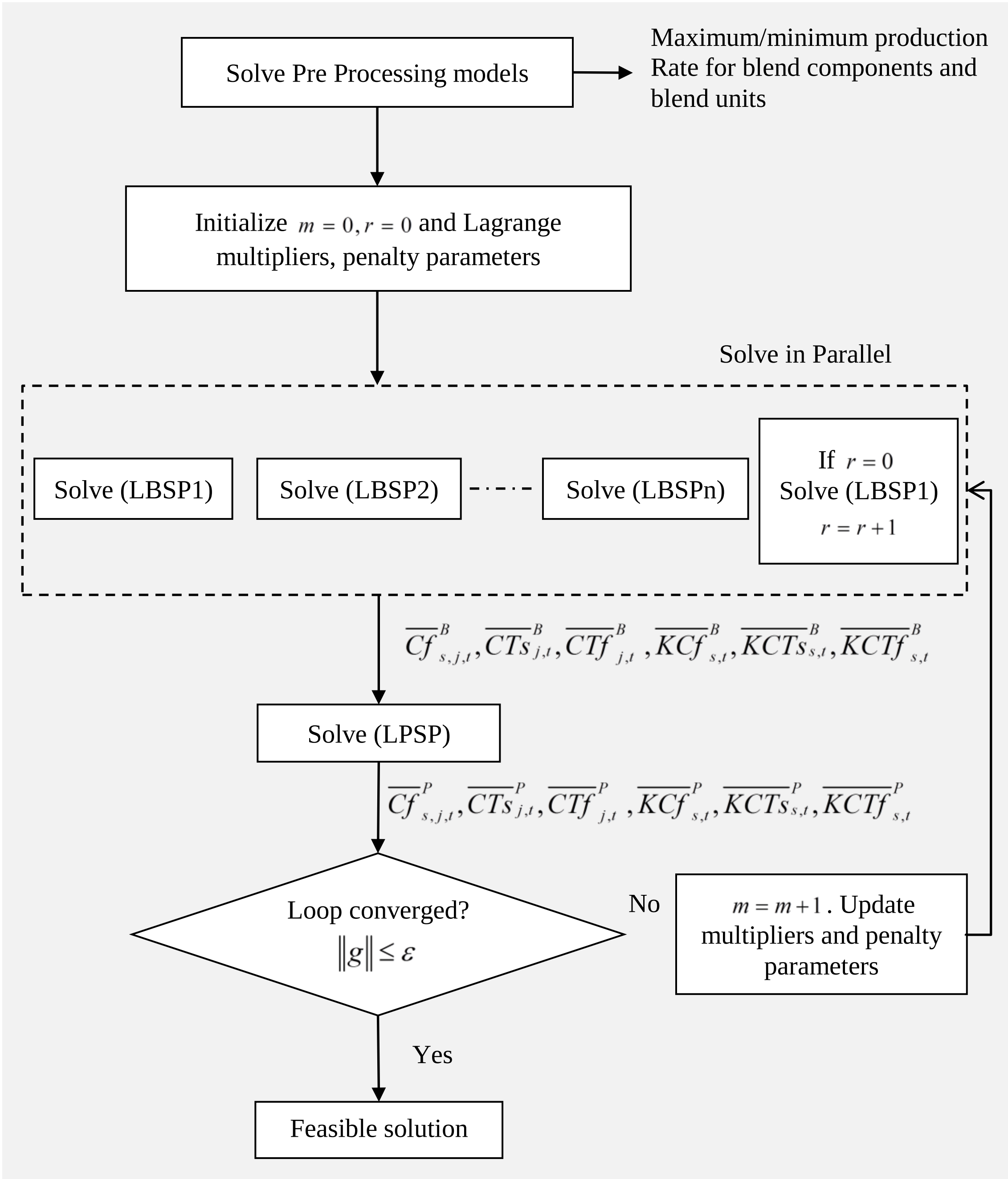

Figure 7.4. Augmented Lagrangian optimization algorithm

7.6. Examples

Three different refinery configurations are used to demonstrate the effectiveness of the proposed algorithm in solving scheduling problems. These refinery configurations are: (1) no blend component tanks, (2) storage options for all blend components, and (3) component tanks for some blend unit.

A real case study provided by Honeywell Process Solution (HPS) presented in Chapter 4 is used to obtain four different examples. The original Honeywell refinery network includes two non-identical blend unit, parallel crude distillation units (CDUs), 13 other production units, 5 intermediate tanks, 9 product tanks, and 2 charging tanks for one of the CDU while other CDU receives crude oil mixture from pipeline. The refinery produces middle distillate products using 8 production units and four finished blend products using two non-identical blend units. A multipurpose Diesel blender produces three different grades of diesel while jet fuel is blended by jet blender. For each blend products, 2 dedicated product tanks are available. In addition to these tanks, a multipurpose product tank is present that services two different grades of diesel product. A minimum of 6 hours maintenance downtime is required when swing tank switches from servicing lower grade product to high grade. This downtime is essential to remove any sulfur or other contamination from swing tank before low sulfur grade product is serviced. Fill-draw-delay for finished blend product tanks is set to 4 hours to let product to settle down and mix properly for certificate analysis. For all production and blend units, minimum of 6 hours uptime (minimum run-length) is required and total available scheduling horizon is 240 hours. Data for the case study is given in Appendix Chapter 3.

Example 1. No component tanks

The case study provided by Honeywell is used.

Example 2. No component tanks and parallel blend units

The original case study (Figure 4.2) is modified by adding a second diesel blend unit alongside the original diesel blender. This new diesel blender produces same three grade products as the original blender and 7 product tanks are available to service diesel blend products produced by both diesel blenders. Table A7-1 gives production rate capacity information about parallel diesel blender.

Example 3. Storage tanks for all blend components

The case study presented in Figure 4.2 is modified by severing connections between production units and blend units (diesel and jet blenders) adding dedicated storage tanks for 6 different blend components and storage capacity data are provided in Table A7-2.

Example 4. Blend component storage and direct flow of component to blend unit

In this example, the refinery has both component storage options and direct flow of component from production units to a blend unit. We include 3 dedicated component tanks for the diesel blender and no blend component storage options for the jet blend unit.

Decomposition:

We decompose the refinery network presented in Figure 4.2 into sub-networks PN and BN by splitting pipeline that transports components from a production unit to a blend unit or component storage tank. Thus, sub-network PN contains charging tanks, CDUs, production units, intermediate tanks and produces blend components, intermediates and intermediate distillate products that are supplied directly to market without storage options whereas sub-network BN includes diesel and jet blenders, component tanks (if any), product tanks, and produces finished blend products diesel and jet fuel. These two different blend products are not stored in same tanks neither do diesel and jet blenders share a common blend component. Thus, we can further decompose sub-network BN into two sub-networks, BN1 and BN2. Network BN1 containing diesel blender, tanks for components blended in diesel blender (if any), and product tanks that service diesel products and corresponding model is (LBSP1). Whereas, sub-network BN2 includes jet blender, components tanks and jet fuel product tanks and relaxed model is (LBSP2). For example 2, two parallel diesel blenders would be present in one network because two parallel blenders share blend components and produce diesel fuels that are stored in same product tanks.

7.6.1. Computational Results

In this section, we investigate the performance of the proposed augmented Lagrangian optimization algorithm. The algorithm is evaluated with three performance measures: number of iterations, and computational time (seconds), and quality of the solution. Data for cost coefficients parameters for objective function is given in Table 7.1. Table A7-3 gives demand for intermediate distillate products. Table A7-4 provides demand for finished blend products, products CARB diesel, EPA diesel, Red-dye diesel, and Jet fuel are referred to as product P1, P2, P3 and P4, respectively. Intermediate demand orders are bound by maximum and minimum amount and can be delivered anytime during the delivery window. Maximum delivery rate, product unloading rate to satisfy demand, is 10 (kbbl/h). All the optimization models are formulated using GAMS, decomposition algorithm implemented in MATLAB R2014a and solved on a Dell® Optiplex 9020 computer with 2.90 GHz Quad Core Intel® Processor i5-4570S, CPU 2.90GHz, and 16.0 GB memory. CPLEX 12.3.0/GAMS 23.7.1 solver is used to solve both MILP and MIQP problems. Optimality tolerance of $1e^{-6}$ % is used for full-scale and pre-processing step models, while for (LBSP) and (LPSP), optimality tolerance of 1% is used and for the algorithm, maximum number of iteration is set to 100.

Table 7.1 Parameters in objective function

Parameter	Value*100	Parameter	Value*100
$C^{1}_{\text{CarbNormal,DieselBlender}}$	10	$C^{7}_{s,s'}$	40 (Unfavorable: 60)
$C^{1}_{\text{EPA Normal,DieselBlender}}$	7	$C^{8}_{s,s'}$	40 (unfavorable: 60)
$C^{1}_{\text{Reddye Normal,DieselBlender}}$	4	C^{9}_{k}	5
$C^{1}_{\text{JetNormal,Jet Blender}}$	5	$C^{10}_{s,k}$	5

C_k^2	0.51 (Swing tank: 1.5)	C_o^{11}	1150
C_k^3	$50 \cdot \left(V_k^{\max}\right)^{-1}$	C_o^{12}	800
C_j^4	50	C_s^{13}	1100
C_k^5	10 (Swing tank: 40)	C_o^{14}	5400
$C_{i,i'}^6$	100 (Unfavorable: 250)	C_o^{15}	3250

Example 1.

This example involves no blend component tanks, 1 multipurpose diesel blender producing three different grades of diesel products that can be stored in 7 different product tanks and 1 jet blender producing jet fuel that can be stored in 2 tanks. One of the diesel product tanks is a swing tank that can store two different type of diesel fuel. For this case study, we select parameters as $c^{16} = (UH)^{-1} e^{-2}$ and $c^{18} = (UH)^{-1} e^{-1}$. These set of parameters drives solution of model (LPSP) for iteration $m = 0$ to that has linking timing variables values much closer to that obtain by (LBSP). This Model sizes for different set of demand examples are shown in Table 7.2. As the number of demand order increase, size of the model increase too and decomposed models have significant less numbers of variables and constraints compared to full-scale model. Furthermore, the complicating variables are expressed in terms of transfer event points, we are able to solve decomposed model with different set of event-points. Full scale model of problem 1 requires minimum of 6 event points to obtain optimal solution while using proposed approach, we are able to solve relaxed diesel blend scheduling model (LBSP1) is solved using 5 event points, relaxed jet blend scheduling model (LBSP2) using 6, and production unit scheduling model (LPSP) using 5 event points. The total numbers of transfer event-points are always equal to the number of event-points n used to solve (LPSP). In Table 7.3, we compare solutions of full space model and proposed decomposition approach for problems of varying demand requirements. In Table 7.3, column '$\lambda g+\sigma\|g\|^2$' represents the value of the augmented and penalty term in the augmented Lagrangian function, '$\|g\|$' represents norm of the consistency constraints.

The computational time to obtain optimal solution increases as the complexity of the model increases and for problem 7, optimality gap is still at 12.91% even after 8 hours. The performance of the algorithm in terms of solution and the quality of the solution is shown in Table 7.3. For all the problems, proposed algorithm provides feasible solution that is closer to the optimal solution reported by full-scale model. For full-scale model that are computationally expensive, proposed algorithm is able to obtain solution in less time. For problem 5, proposed approach is able to report feasible solution of 68717 in 2755 seconds while optimal solution of 67286 is obtained by full scale model in 5115 seconds. The solution procedure for problem 5 is shown in Figure 7.5. The solution time for algorithm is driven by relaxed blend scheduling problem (LBSP). More specially, relaxed diesel blending (LBSP1) is the most difficult to solve due to presence of multipurpose blend unit and multipurpose product tanks.

Table 7.2 Model statistics for Example 1

Ex.	**# Demand Orders**	**Full Scale Model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.	**Augmented Lagrangian Algorithm** **LBSP model** Event pt. n Int./Cont. variables (Constraints) Nonzero Elem.		**Augmented Lagrangian Algorithm** **LPSP model** Event pt. n Int./Cont. var. (Constraints) Nonzero Elem.
			LBSP1	**LBSP2**	
1	4	6 434/3197 (7748)	5 127/742 (2594)	6 37/243 (985)	5 235/1610 (4859)

		28085	9794	3556	17498
2	4	5 360/2663 (6392) 22608	5 116/736 (2545) 9584	5 30/200 (752) 2571	4 188/1288 (3727) 13045
3	4	4 286/2129 (5050) 17461	4 100/592 (1967) 7096	4 26/164 (587) 1926	4 188/1286 (3719) 13022
4	6	4 301/2217 (5339) 18547	4 109/644 (2134) 7718	4 32/200 (689) 2284	4 188/1286 (3721) 13021
5	8	6 484/3453 (8653) 31676	6 173/1039 (3739) 14750	6 57/347 (1335) 4888	5 235/1611 (4862) 17508
6	8	6 484/3453 (8643) 31586	6 173/1039 (3729) 14661	6 57/347 (1335) 4888	5 235/1611 (4862) 17508
7	13	7 640/4383 (11642) 44572	7 266/1511 (5718) 23720	7 80/467 (1930) 7518	6 282/1936 (6092) 22632

Table 7.3 Numerical results for Example 1

Ex	**Full Scale model**			**Augmented Lagrangian Algorithm** $\sigma_{Cf}^{(0)} = 0.2$, $\sigma_{CTs}^{(0)} = 0.3$, $\sigma_{CTf}^{(0)} = 0.2$, $\gamma = 0.37$, $\beta = 2.15$				
	CPU(sec)	Objective $\left(z \times e^{-2}\right)$	Gap (%)	Iter. m	Time (sec)	$f \times e^{-2}$	$\lambda g + \sigma \|g\|^2$	$\|g\|$
1	608.73	501.89	0.00	10	368.02	502.35	11.90	0.70
2	73.29	428.25	0.00	12	525.29	428.19	541.73	0.87
3	16.35	788.98	0.00	9	138.95	791.77	-16.23	0.40
4	36.12	735.16	0.00	15	195.30	747.36	-2.98	0.34
5	5114.95	672.86	0.00	14	2755.25	687.17	1693.37	0.86
6	23085.76	866.43	0.00	8	3001.96	869.75	37.65	0.76
7	28800	695.79	12.91	10	14720.16	695.62	122.12	0.90

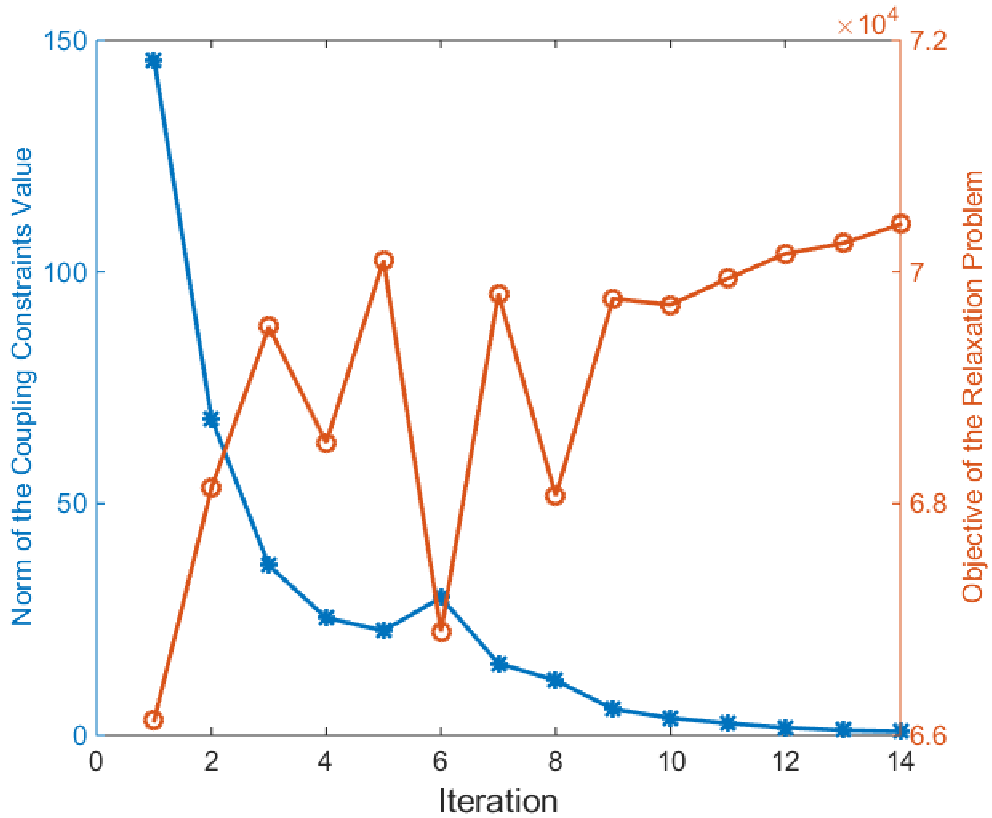

Figure 7.5 Solution procedure for Example 1, problem 5.

Example 2.

Refinery configuration with parallel blend unit and no component flow is examined to check efficacy of the proposed decomposition approach. Parameters for model (LPSP) objective function are chosen to be $c^{16} = (UH)^{-1} e^{-2}$, $c^{18} = (UH)^{-1} e^{-1}$, and $c^{20} = 0.75 c_j^1$. Parallel diesel blenders that can produce three different grade of diesel product are present thus compared to Example 1, this refinery configuration require less number of event points for blend scheduling model (LBSP) to obtain optimal solution that satisfy demand orders for finished blend products. Decomposition provides smaller scheduling problem than full scale scheduling problem as shown in Table 7.4. Proposed decomposition algorithm is able to provide a good feasible solution and in reasonable solution time, as reported in Table 7.5. Solution procedure for problem 5 is shown in

Figure 7.6. The solution time in ALO-DQA algorithm is mainly spent solving relaxed problem corresponding to multipurpose diesel blend units.

Table 7.4 Model statistics for Example 2

Ex.	**# Orders**	**Full Scale Model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.	**Augmented Lagrangian Algorithm**		
			LBSP model Event pt. Int./Cont. variables (Constraints) Nonzero Elem.		**LPSP model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.
			LBSP1	**LBSP2**	
1	4	4 330/2425 (6319) 22522	4 122/882 (3465) 12906	4 23/159 (550) 1788	3 150/1010 (2928) 10218
2	4	4 330/2425 (6319) 22522	4 122/882 (3465) 12906	4 23/159 (550) 1788	3 150/1010 (2928) 10218
3	4	3 245/1817 (4655) 16274	3 106/668 (2512) 9059	3 19/124 (408) 1280	3 150/1007 (2929) 10205
4	6	3 255/1885 (4883)	3 112/708 (2647)	3 23/152 (483)	3 150/1007 (2929)

		17107	9558	1539	10202
5	8	5 455/3249 (8821) 32386	5 184/1238 (5084) 19672	5 46/289 (1040) 3636	4 200/1347 (4139) 14870
6	8	4 360/2601 (6953) 24884	4 140/986 (3835) 14237	4 35/231 (776) 2598	3 150/1010 (2936) 10230
7	13	5 503/3509 (9941) 36900	5 224/1454 (5975) 23174	5 54/333 (1223) 4345	4 200/1347 (4139) 14870

Table 7.5 Numerical results for Example 2

Ex	**Full Scale model**			**Augmented Lagrangian Algorithm** $\sigma_{Cf}^{(0)} = 0.3$, $\sigma_{CTs}^{(0)} = 0.3$, $\sigma_{CTf}^{(0)} = 0.3$, $\gamma = 0.30$, $\beta = 2.20$				
	CPU(sec)	Objective $\left(z \times e^{-2}\right)$	Gap (%)	Iter. m	Time (sec)	$f \times e^{-2}$	$\lambda g + \sigma \lVert g \rVert^2$	$\lVert g \rVert$
1	559.25	464.88	0.00	8	446.08	474.23	-7.65	0.379
2	82.62	382.35	0.00	10	539.03	413.51	92.67	0.568
3	39.58	629.05	0.00	9	263.35	630.02	-2.50	0.794
4	16.30	579.62	0.00	11	576.03	584.46	-14.43	0.278
5	19466.87	661.53	0.00	16	9700.18	705.76	10580.87	0.971

6	12872.25	756.43	0.00	7	2251.59	767.34	-2.49	0.252
7	28800	683.57	13.24	7	13300.39	692.36	-15.76	0.659

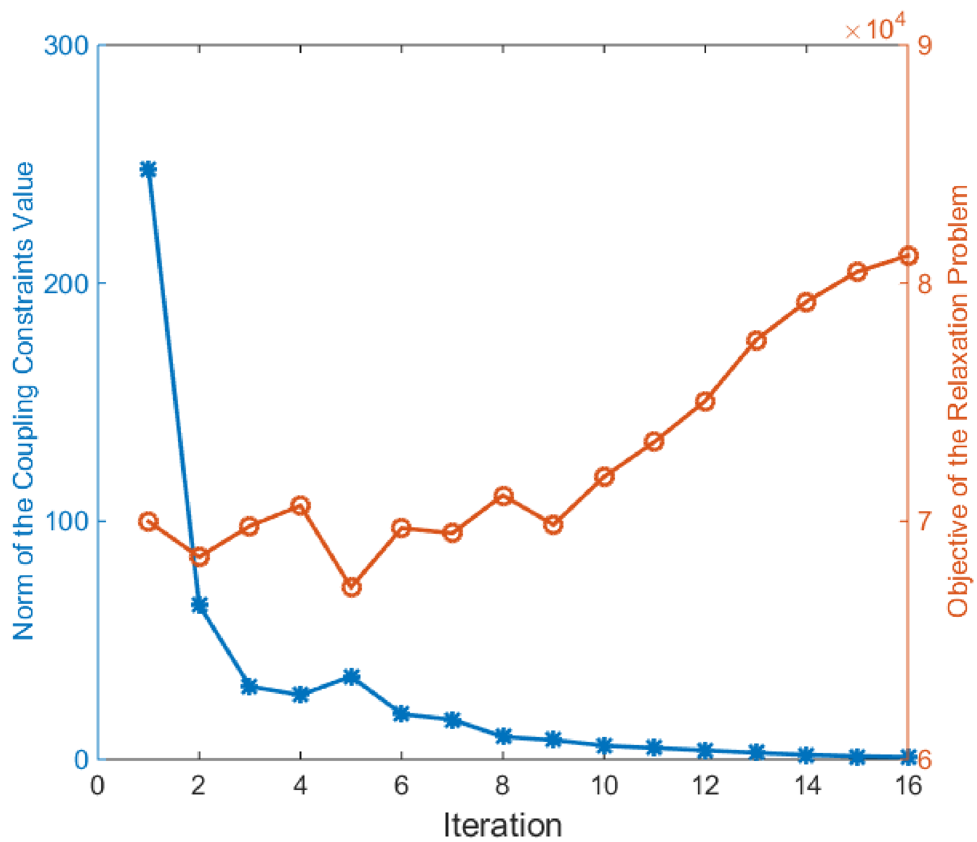

Figure 7.6 Solution procedure for Example 2, problem 5

Example 3.

Refinery configuration with dedicated component tanks is studied in this example. We include 1 storage tank for each blend component and eliminate all direct streams from production units to blenders. For all problems, parameters value are set at $c^{17} = 0.01$, $c^{19} = 0.5$. Model statistics for this example is shown in Table 7.6 and computational results are given in Table 7.7. Augmented Lagrangian algorithm converges to a feasible solution for each problem within 20 iterations and proposed multipliers and penalty parameters gives a solution that is closer to original full-scale model optimal solution. However, for problem 1, the solution obtain using decomposition algorithm is far worse than optimal solution, and this is due to the augmented Lagrangian parameters and penalty. For problem 1, the algorithm converges to a solution that has high task changeovers cost. Convergence profile for problem 5 is given in Figure 7.7. Similar to other example, relaxed diesel blend scheduling (LBSP1) is most difficult to solve and other two problems take only few seconds.

Table 7.6 Model statistics for Example 3

Ex.	**# Orders**	**Full Scale Model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.	**Augmented Lagrangian Algorithm**		
			LBSP model Event pt. Int./Cont. variables (Constraints) Nonzero Elem.		**LPSP model** Event pt. Int./Cont. var. (Constraints) Nonzero Elem.
			LBSP1	**LBSP2**	
1	4	5 455/2978 (7230) 24793	5 169/927 (3464) 12582	5 23/159 (550) 1788	5 150/1010 (2928) 10218
2	4	5 455/2978 (7230) 24793	5 122/882 (3465) 12906	4 23/159 (550) 1788	4 150/1010 (2928) 10218
3	4	4 362/2381 (5715) 19193	4 106/668 (2512) 9059	4 19/124 (408) 1280	4 150/1007 (2929) 10205

4	6	4 377/2469 (6004) 20279	4 112/708 (2647) 9558	4 23/152 (483) 1539	4 150/1007 (2929) 10202
5	8	6 598/3831 (9664) 34314	6 184/1238 (5084) 19672	6 46/289 (1040) 3636	6 200/1347 (4139) 14870
6	8	6 598/3831 (9654) 34224	6 140/986 (3835) 14237	6 35/231 (776) 2598	5 150/1010 (2936) 10230

Table 7.7 Numerical results for Example 3

Ex	**Full Scale model**			**Augmented Lagrangian Algorithm** $\sigma_{KCf}^{(0)} = 0.05$, $\sigma_{KCTs}^{(0)} = 0.05$, $\sigma_{KCTf}^{(0)} = 0.05$, $\gamma = 0.25$, $\beta = 2.20$				
	CPU(sec)	Objective $\left(z \times e^{-2}\right)$	Gap (%)	Iter. m	Time (sec)	$f \times e^{-2}$	$\lambda g + \sigma \Vert g \Vert^2$	$\Vert g \Vert$
1	266.31	475.54	0.00	8	1063.97	1743.94	7.96	0.109
2	254.30	421.41	0.00	11	464.76	565.70	-296.52	0.428
3	56.71	730.47	0.00	9	154.27	738.25	0.09	0.202
4	147.78	711.69	0.00	13	239.89	726.78	82.14	0.454
5	28800	654.61	1.06	16	3766.46	658.51	1199.03	0.952
6	28800	866.43	25.18	7	21496.02	871.84	-29.66	0.956

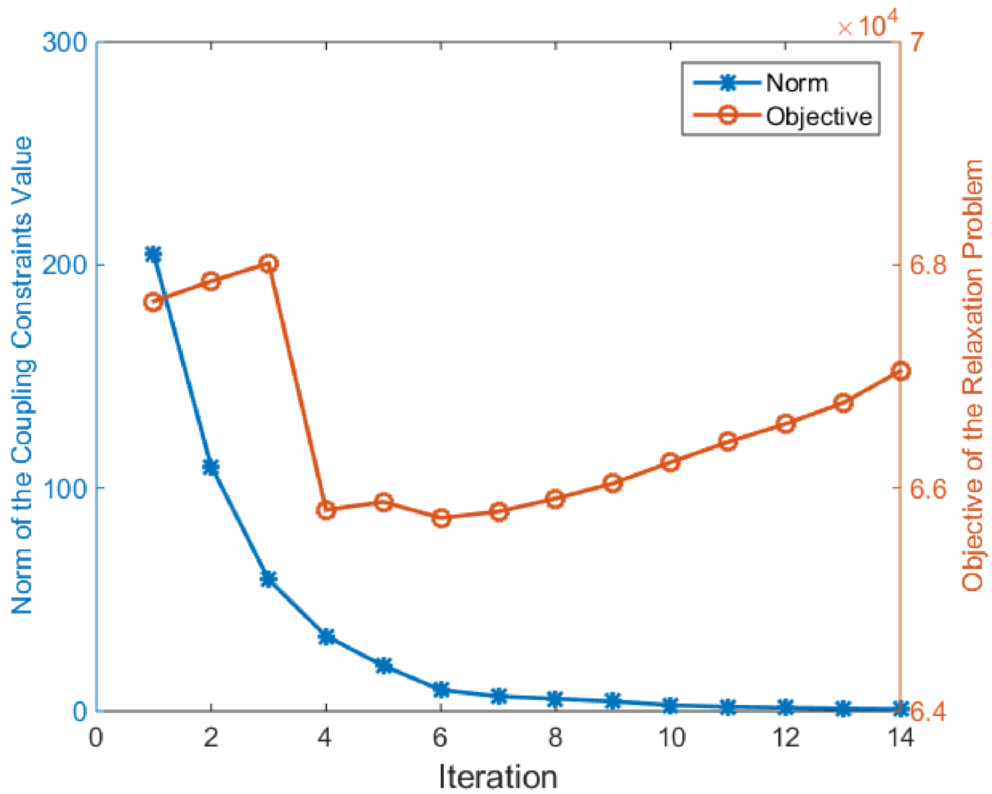

Figure 7.7 Solution procedure for Example 3, problem 5.

Example 4.

In this example, we test the proposed algorithm for a refinery that has blend component tanks and direct flow to blend units. Model statistics and computational results are shown in Table 7.8 and Table 7.9 for problem 5.

Table 7.8 Model statistics for Example 5

	Full scale Model	**Augmented Lagrangian Optimization Approach**		
		LBSP1	**LBSP2**	**LPSP**
Event points	6	6	6	5
Binary variables	538	232	57	220
Continuous	3633	1315	347	1664

Variables				
Constraints	9097	4863	1335	5021
Nonzero Elements	32774	18599	4888	17791

Table 7.9 Numerical results for Example 5

	Full scale Model	**Augmented Lagrangian Optimization Approach**
CPU time (Sec)	28800	1556
Objective	66895	69938
Gap (%)	10.67	
Iterations (m)		8
$\lambda g + \sigma \lVert g \rVert^2$		359.97
$\lVert g \rVert$		0.862

Full-scale model is not able to obtain an optimal solution even after 8 hours, whereas proposed approach provides a feasible solution in 1556 seconds. Parameters present in model (LPSP) objective function has value of $c^{16} = (UH)^{-1} e^{-2}$, $c^{17} = 0.01$, $c^{18} = (UH)^{-1} e^{-1}$, and $c^{19} = 0.5$. Solution procedure for problem 5 is shown in Figure 7.8.

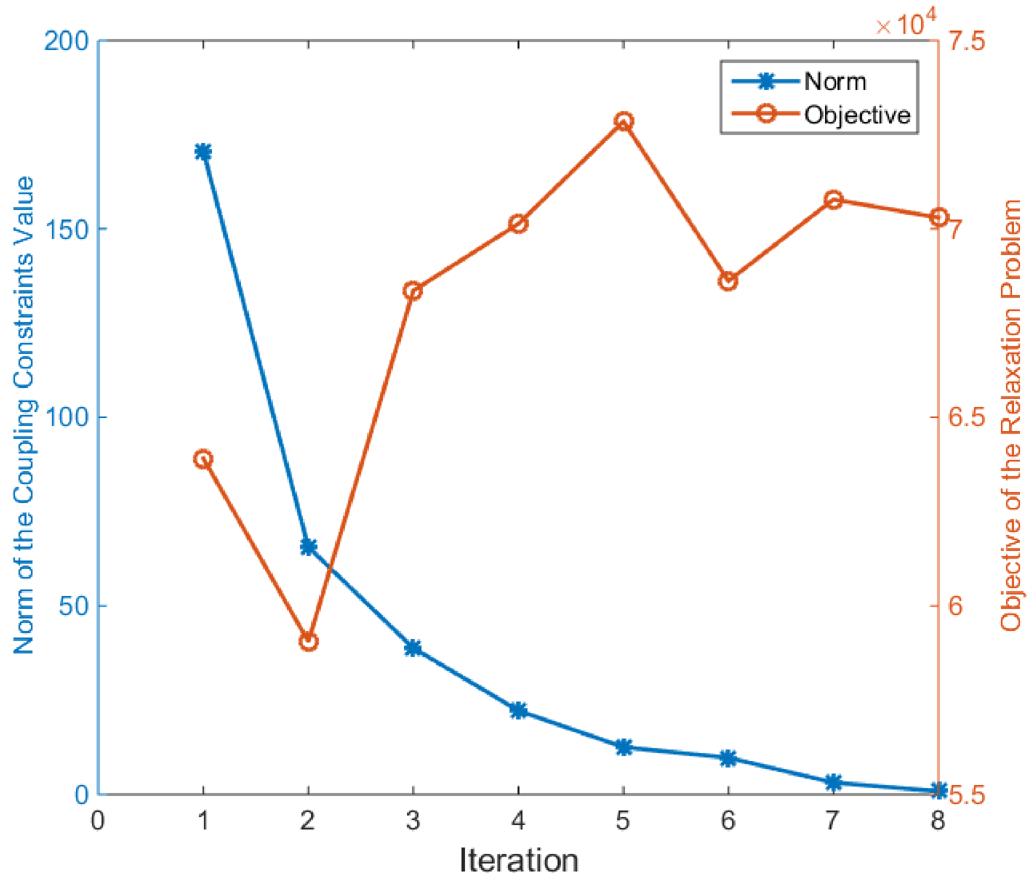

Figure 7.8 Solution procedure for Example 4, problem 5.

From above results for all four examples, it can be observed that the augmented Lagrangian algorithm converges to a feasible solution of the full-scale problem because the linking variables error norm value always convergences to zero as augmented penalty term $\sigma \to \infty$. Quality of the feasible solution obtain and solution time is not only affected by augmented Lagrangian optimization parameters but also by parameters $c^{16}, c^{17}, c^{18}, c^{19}$, and c^{20} in some extant.

7.7. Summary

We have applied Augmented Lagrangian approach to refinery operations scheduling problem by decomposing the problem into production unit operations and finished blend product and delivery operations. Continuous time formulation based on event-points is used in this work and complicating variables and coupling constraints for decomposed problems are expressed in terms of novel transfer event-points. To improve the performance of the algorithm, a preprocessing step and constraints to incorporate information lost during decomposition are proposed. Proposed approach is shown to effectively address four different refinery network configurations.

Main advantages of the proposed decomposition approach are: (a) decomposed problems can be solved using different set of event-points n because complicating variables and coupling variables are expressed in terms of transfer event-points t; (b) when blending operations network can be decomposed further, this method allows for parallel implementation of blend scheduling problems; (c) the convergence of the algorithm to a feasible solution is guaranteed. Furthermore, this approach can be extended to include crude-oil loading and unloading operations so entire refinery operations scheduling problem can be integrated.

Nomenclature

Indices

bn	Blending operations sub-network
i	Tasks
j	Production units
k	Storage units
n	Event-points
m	Lagrangian algorithm iteration counter
o	Product order
p	Properties
s	States
t	Transfer event-points

Sets

BN	All decomposed sub-networks belonging to blending operations
I_j	Tasks which can be performed in unit j
I_s^c / I_s^p	Tasks which can consume/produce material s
J	All Production units
J^{ads}	Blend units that receive at least all components directly from production units
J^{BU}	Blend units
J^{cont}	All production units and blend units that receive at least one component directly from production unit operations
J^{ds}	Blend units that receive at least one component directly from production units
J^{PU}	Production units that belong to production unit operations
J_s^c / J_s^p	Units that consume/produce material s
J_i	Units which are suitable for performing task i
J^h	Units that can produce all the same products as some other unit in the refinery
J^m	Units which are suitable for performing multiple tasks
J_j^{seq}	Units that follow unit j (no storage in between)
J_k^{kp} / J_k^{pk}	Units that consume/produce material s stored in tank k
J^P	Units that produce products
K	Storage units
K^h	tanks that can store the same products as some other tank in the refinery
K_j^{kp} / K_j^{pk}	Tanks that store material consumed/produce by unit j
K^m	Multipurpose tanks that can store multiple materials
K_p	Tanks that can store final products
K_s	Tanks that can store material s

N	Event-point within the time horizon
O_s	Orders for finished blend product s that is stored in tanks
P	Product Properties
S	States
S_A	Group A products, stored in tanks
S_B	Group B products, not stored in tanks
S_{bc}	Blend components states
S^{bn}	States that belong to sub-network bn
S_k	States that can be stored in tank k
S_i^c / S_i^p	States that can be consumed/produced by task i
S_j^c / S_j^p	States that can be consumed/produced by unit j

Parameters

$cont_{s,s'}$	1 if components s and s' are produced by units in PN that are interconnected
$D_{o,s}^+, D_{o,s}^-$	Demand limit requirement for order o and product s that is stored in tank
max_rate_s	Maximum rate of production of material s
$ot_{o,i}$	1 if task i is performed when order o of PUSP is processed by BSP.
r_s	Demand of the final product s at the end of the time horizon
$RB_j^{\min}$ / $RB_j^{\max}$	minimum and maximum production rate of a blend unit j
$RP_j^{\max}$	Maximum rate material is supplied by (PSP) to blend unit j
$RP_s^{\max}$	Maximum rate blend component s is supplied by (PSP) to blend unit j
$R_{i,j}^{\min}$ / $R_{i,j}^{\max}$	Minimum/ maximum rate of material be processed by task i in unit j
RL_i	Minimum run length for task i
$RU_k^{\min}$ / $RU_k^{\max}$	Minimum/maximum rate of product unloading at tank k

$sto_{s,k}$	Amount of state *s* that is present at the beginning of the time horizon in *k*
ta_k	Fill draw delay for product tank *k*
UH	Available time horizon
$V_k^{\max}$	Maximum available storage capacity of storage tank *k*
V_k^{heel}	Maximum heel available for storage tank *k*
$yo_{s,k}$	1 if the material *s* is present at the beginning of the time horizon in k
$\rho_{s,i}^{\min} / \rho_{s,i}^{\max}$	Proportion of state *s* produced/consumed by task *i*
$\rho b_{s,j}^{\min} / \rho b_{s,j}^{\max}$	Minimum/maximum proportion of blend component *s* consumed by blend unit *j*
φ_s	Quality index for finished blend product, higher index means better quality

Variables

Binary Variables

$a_{j',j,n}$	Assigns the materials flow to blend unit *j* from unit j' at event-point *n*
$wv_{i,j,n}$	Assignment of task *i* in unit *j* at event-point *n*
$in_{s,j,k,n}$	Assigns the material flow of *s* into storage tank *k* from unit *j* at point *n*
$l_{k,o,n}$	Assigns the starting of product flow out of product tank *k* to satisfy order *o* at event-point *n*
$out_{s,k,j,n}$	Assigns the material flow of *s* out of storage tank *k* into unit *j* at point *n*
$y_{s,k,n}$	Denotes that material *s* is stored in tank k at event-point n

0-1 continuous variables

$\alpha_{j,n}$	For unit *j*, 1 if the unit becomes active for very first time at event-point *n*
$\beta_{k,n}$	For tank k, 1 if the tank becomes active for very first time at event-point *n*
$\eta_{s,s',k,n}$	Continuous 0-1 variable, 1 if material in tank *k* switchover service from s at

	event-point n to s' at later event-point
$\eta o_{s,s',k}$	Continuous 0-1 variable, 1 if material in tank k switchover service from s to s'
$\chi_{i,i',j,n}$	Continuous 0-1 variable, 1 if task at unit j changes from i at event-point n to i'at later event-point.
$x_{j,n,t}$ / $xk_{s,n,t}$	Cont. 0-1 variable denotes transfer event-point t corresponds to event-point n

Positive variables

$bp_{s,i,j,n}$	Amount of material s produced task i in unit j at event-point n
$bc_{s,i,j,n}$	Amount of material s undertaking task i in unit j at event-point n
$Cf_{s,j,t}$ / $KCf_{s,t}$	Amount of blend component s transferred between BN and PN at event-point t
dg_o^l	Minimum demand quantity give-away term for order o
dg_o^u	Maximum demand quantity give-away term for order o
H	time horizon used for production tasks
$JJf_{s,j,j',n}$	Flow of state s from unit j to consecutive unit j' for consumption at point n
$Kif_{s,j,k,n}$	Flow of material s from unit j to storage tank k event-point n
$Kof_{s,k,j,n}$	Flow of material s from storage tank k to unit j at point n
$Lf_{o,k,n}$	Flow of final product for order o from storage tank k at event-point n
$mh_{s,k,n}$	Maximum heel give-away term for product tanks
rg_s	Minimum demand quantity give-away term for Group B product s
$Rif_{s,k,n}$	Flow of raw material to storage tank k event-point n
$st_{s,k,n}$	Amount of state s present in storage tank k at event-point n
$std_{s,s',k,n}$	Amount of state s that is downgraded to state s' in storage tank k at event-point n
$Tearly_o$	Early fulfillment of order o than required

$Tf_{i,j,n}$	Time that task *i* finishes in unit *j* at event-point *n*
$CTf_{j,t}$ / $KCTf_{s,t}$	Finish time for components transfer between BN and PN at time-point *t*
$Tlate_o$	Late fulfillment of order *o* than required
$Tos_{k,o,n}$	Time that material starts to flow from tank *k* for order o at event-point *n*
$Tof_{k,o,n}$	Time that material finishes to flow from tank *k* for order o at event-point *n*
$Ts_{i,j,n}$	Time that task *i* starts in unit *j* at event-point *n*
$CTs_{j,t}$ / $KCTs_{s,t}$	Start time for components transfer at time-point *t*
$Tsf_{j,k,n}$	Time that material finishes to flow from unit *j* to tank *k* at event-point *n*
$Tsf_{k,j,n}$	Time that material finishes to flow from tank *k* to unit *j* at event-point *n*
$Tss_{j,k,n}$	Time that material starts to flow from unit *j* to storage tank *k*
$Tss_{k,j,n}$	Time that material starts to flow from tank *k* to unit *j* at event-point *n*
$Uif_{s,j,n}$	Flow of raw material *s* to production unit *j* at point *n*
$Uofb_{s,j,j',n}$	Flow of component *s* from production unit *j* to blend unit j' at point *n*
$Uof_{s,j,n}$	Flow of product material *s* from unit *j* at point *n*
$UUf_{j,s,v,n}$	Amount of state *s* received by unit *j* at period *v* is processed at event *n*
z	Objective value of full-scale integrated model

Acknowledgement of Prior Works

All of the work presented in this dissertation represents original research by the author. However some of the concepts presented in Chapters 2 to Chapter 6 have previously appeared in published works, for which the author of this dissertation is also the first author. The author of this dissertation is also first author on these prior works, which include:

- Shah, N. K., & Ierapetritou, M. G. (In Process). Lagrangian Decomposition Approach to Scheduling of a Large-Scale Oil-Refinery Operations.
- Shah, N. K., & Ierapetritou, M. G. (In Process). Efficient Heuristic Algorithm Strategy for Short-Term Scheduling of Refinery Operations.
- Shah, N. K., & Ierapetritou, M. G. (2012). Integrated production planning and scheduling optimization of multisite, multiproduct process industry. Computers and Chemical Engineering, 37, 214-226.
- Shah, N. K., Li, Z., & Ierapetritou, M. G. (2011). Petroleum Refining Operations: Key Issues, Advances, and Opportunities. Industrial & Engineering Chemistry Research, 50, 1161-1170.
- Shah, N. K., & Ierapetritou, M. G. (2011). Short-term scheduling of a large-scale oil-refinery operations: Incorporating logistics details. AIChE Journal, 57, 1570-1584.
- Shah, N., Saharidis, G. K. D., Jia, Z., & Ierapetritou, M. G. (2009). Centralized-decentralized optimization for refinery scheduling. Computers & Chemical Engineering, 33, 2091-2105.

Appendix

Appendix Chapter 2

Table A2-1 Example 1: Process data for production sites

Production Site	**Unit**	**Capacity**	**Suitability**	**Processing Time**
S1	Heater	100	Heating	1.0
	Reactor 1	50	Reactions 1,2,3	2.0,2.0,1.0
	Reactor 2	80	Reactions 1,2,3	2.0,2.0,1.0
	Sill	200	Separation	1 for Product 2 2 for IntAB
S2	Heater	150	Heating	1.0
	Reactor 1	100	Reactions 1,2,3	2.0,2.0,1.0
	Reactor 2	80	Reactions 1,2,3	2.0,2.0,1.0
	Sill	300	Separation	1 for Product 2 2 for IntAB
S3	Heater	100	Heating	1.0
	Reactor 1	75	Reactions 1,2,3	2.0,2.0,1.0
	Reactor 2	100	Reactions 1,2,3	2.0,2.0,1.0
	Sill	150	Separation	1 for Product 2 2 for IntAB
Production Site	**State**		**Storage Capacity**	**Initial Amount**
S1	Feed A		100000	100000
	Feed B		100000	100000

	Feed C		100000	100000
	Hot A		100	0.0
	Int AB		200	0.0
	Int BC		150	0.0
	Impure		200	0.0
	Product 1		400	0.0
	Product 2		375	0.0
S2	Feed A		100000	100000
	Feed B		100000	100000
	Feed C		100000	100000
	Hot A		130	0.0
	Int AB		250	0.0
	Int BC		150	0.0
	Impure		300	0.0
	Product 1		300	0.0
	Product 2		425	0.0
S3	Feed A		100000	100000
	Feed B		100000	100000
	Feed C		100000	100000
	Hot A		115	0.0
	Int AB		250	0.0
	Int BC		150	0.0
	Impure		300	0.0
	Product 1		350	0.0
	Product 2		400	0.0

Table A2-2 Production and Inventory cost data

Production Sites		Fixed Cost	Variable Costs		Inventory Cost
S1	Heating	150	1.0		
	Reactions 1,2,3	100	0.5	P1	10
				P2	10
	Separation	150	1.0		
S2	Heating	175	1.5		
	Reactions 1,2,3	125	0.75	P1	15
				P2	15
	Separation	175	1.5		
S3	Heating	100	0.8	P1	8
				P2	8

Table A2-3 Transportation and Backorder cost data

Market		Transportation Cost		Backorder Cost
M1	S1	5	P1	100
	S2	8	P2	100
	S3	10		
M2	S1	10	P1	150
	S2	5	P2	150
	S3	8		
M3	S1	8	P1	75
	S2	10	P2	75
	S3	8		

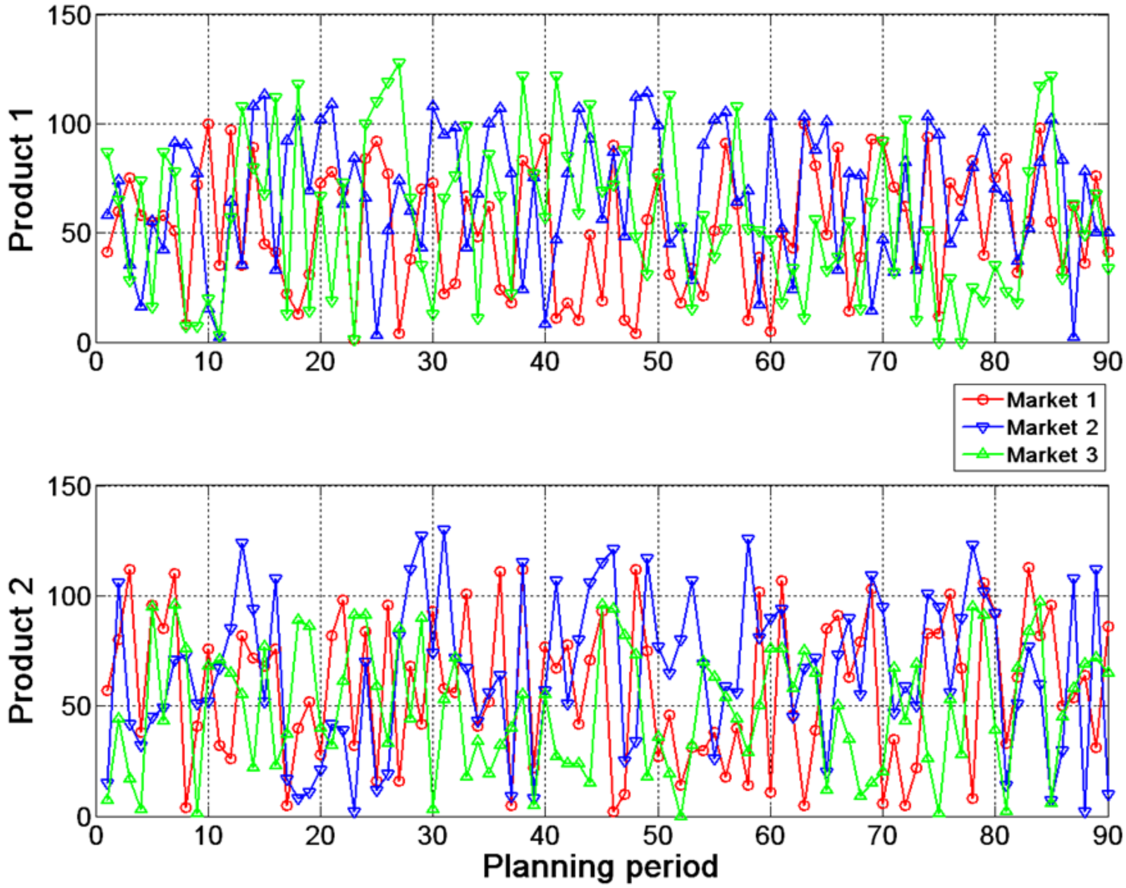

Figure A2-1 Demand data for Example 1

Appendix Chapter 3

Table A3-1 Demand in Thousand barrels

Final Products	Scenario 1				Scenario 2			
	Case 1	Case 2	Case 3	Case 4	Case 1	Case 2	Case 3	Case 4
FCC gas	10	50	35	75	75	150	200	200
Coke	10	35	10	75	75	75	100	150
Carb diesel	10	100	50	100	100	300	350	500
EPA diesel	10	100	50	150	150	250	250	250
Red dye diesel	10	100	50	150	150	250	300	250

Table A3-2 Production rates in thousand barrels/hour

Unit	Task	R^{min}	R^{max}
4CU	4CU Normal	0.1	7.292
2CU	2CU Normal	0.1	4.167
Vacuum Tower	Vacuum Normal	0.1	5.708
Coker	Coker Normal	0.1	2.75
FCC HDS	FCCHDS Normal	0.1	3.00
FCC	Maxdistillation Mode	0.1	2.708
FCC	Maxgasoline Mode	0.1	2.708

Diesel	Diesel Normal	0.1	2.5
Isomax	Isomax Normal	0.1	2.042
Diesel Blemder	Carb Diesel	0.1	54.167
Diesel Blender	EPA Diesel	0.1	54.167
Diesel Blender	RedDye Diesel	0.1	54.167

Table A3-3 Recipe data

Task	State Produced	ρ^P	State Consumed	ρ^c
4 CU Normal	Diesel Resid	0.691 0.309	ANS1	1
2 CU Normal	Diesel Resid	0.284 0.716	SJV Crude	1
Vacuum Tower Normal	Vacuum Resid Heavy Gasoil Light Gasoil	0.334 0.333 0.333	Resid	1
Coker Normal	Coke Heavy Gasoil Light Gasoil	0.640 0.180 0.180	Vacuum Resid	1
FCC HDS Normal	FCC HDS	1	Heavy Gasoil	1
Maxdistillation Mode	FCC Gas LCO	0.50 0.50	FCC HDS	1

Maxgosoline Mode	FCC Gas LCO	0.50 0.50	FCC HDS	1
Diesel Normal	HDS Diesel	1	Diesel	1
Isomax Normal	Hydro Diesel	1	Light Gasoil	1
Carb Diesel Normal	Carb Diesel	1	HDS Diesel Hydro Diesel LCO	0.20 0.25 0.55
EPA Diesel Normal	EPA Diesel	1	HDS Diesel Hydro Diesel LCO	0.25 0.15 0.60
Red Dye Diesel Normal	Red Dye Diesel	1	HDS Diesel Hydro Diesel LCO	0.23 0.12 0.65

Table A3-4 Storage tank capacity data in thousand barrels

Storage Tank	Material Stored	Capacity	Initial Amount
ANS Feed Tank 1	ANS	750	10
ANS Feed Tank 2	ANS	750	10
Coker Feed Tank	Vacuum Resid	500	0
FCC HDS Feed Tank	Heavy Gasoil	500	0
Diesel HDS Feed Tank	Diesel	500	0

Isomax Feed Tank	Light Gasoil	500	0
Carb Diesel Tank 1	Carb Diesel	200	0
Carb Diesel Tank 2	Carb Diesel	200	0
Diesel Swing Tank	Carb Diesel	200	0
EPA Diesel Tank 1	EPA Diesel	125	0
EPA Diesel Tank 2	EPA Diesel	125	0
RedDye Diesel Tank1	RedDye Diesel	150	0
RedDye Diesel Tank 2	RedDye Diesel	150	0

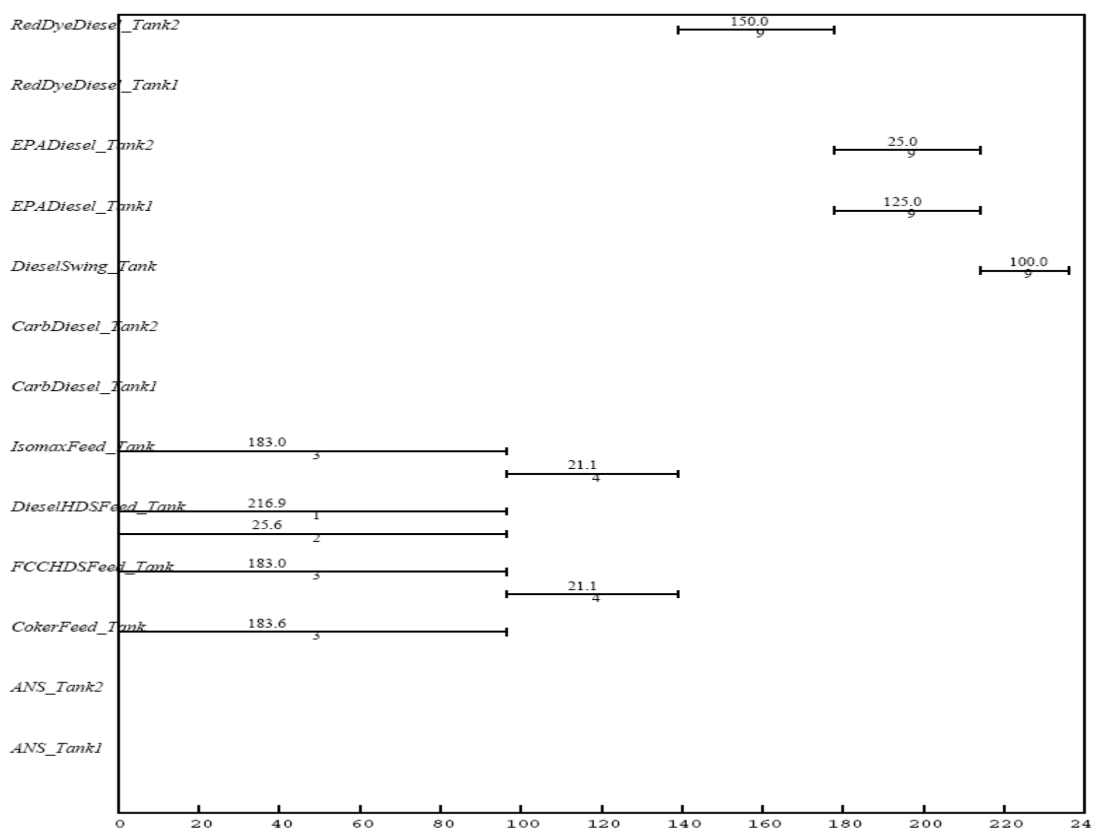

Figure A3-1 Gantt chart of tank loading schedule for example 1, Centralized system, scenario 1

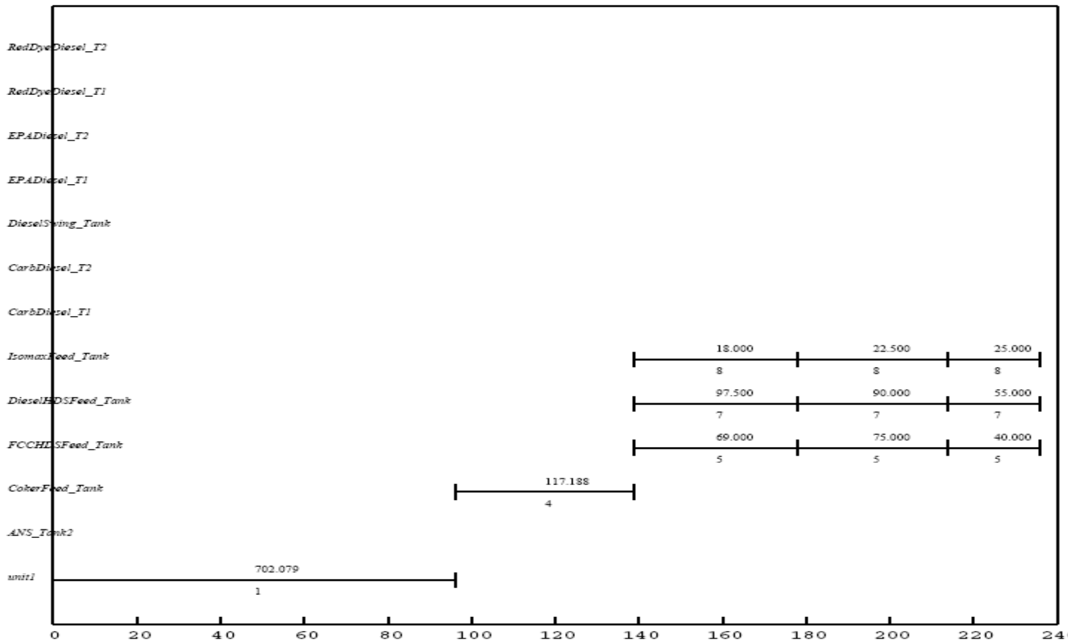

Figure A3-2 Gantt chart of the tank unloading schedule for example 1, Centralized system, scenario 1

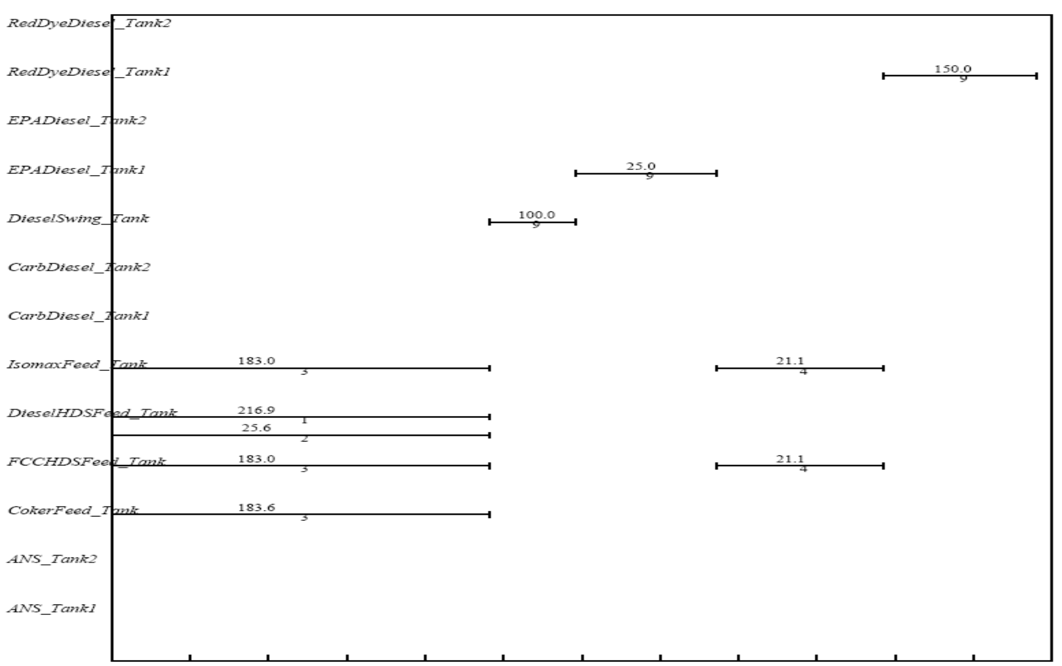

Figure A3-3 Gantt chart of tank loading schedule for example 1, Decentralized system, scenario 1

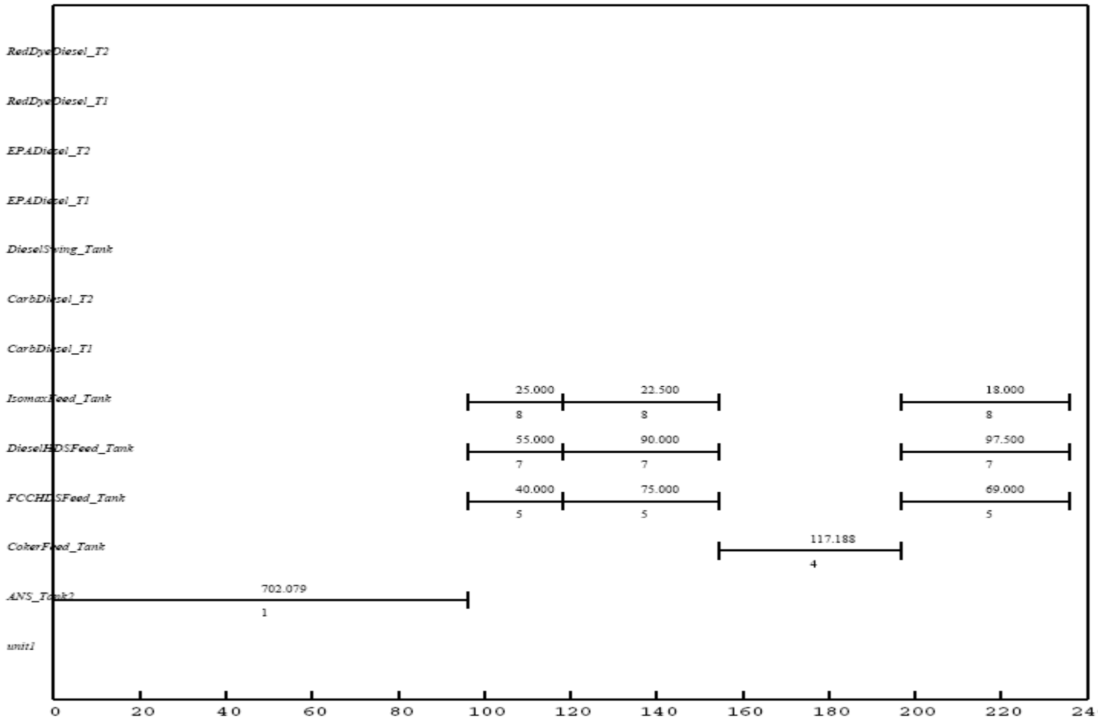

Figure A3-4 Gantt chart of the tank unloading schedule for example 1, Decentralized system, scenario 1

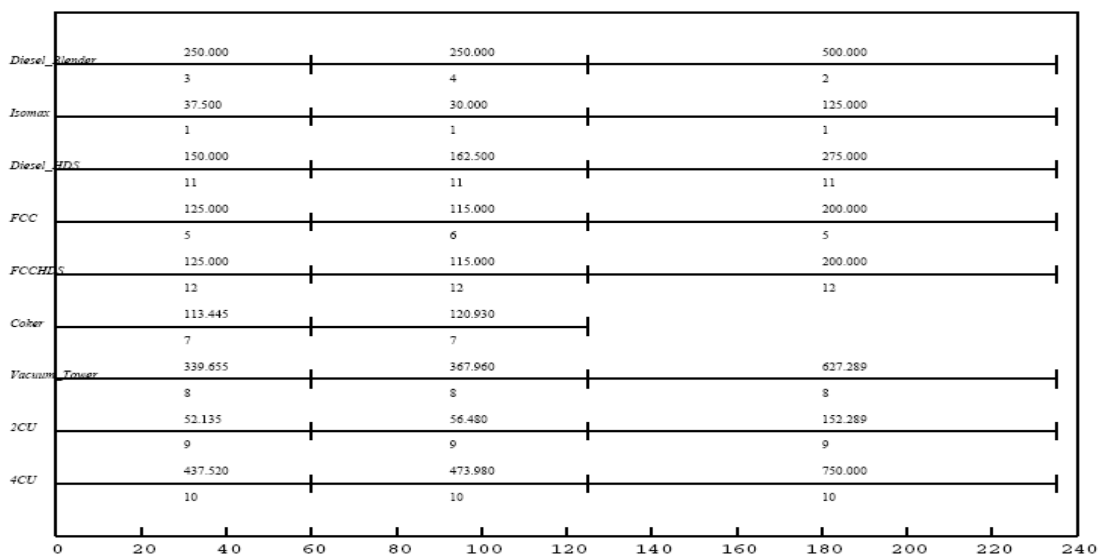

Figure A3-5 Gantt chart of the operation schedule for example 1, Centralized system, scenario 2

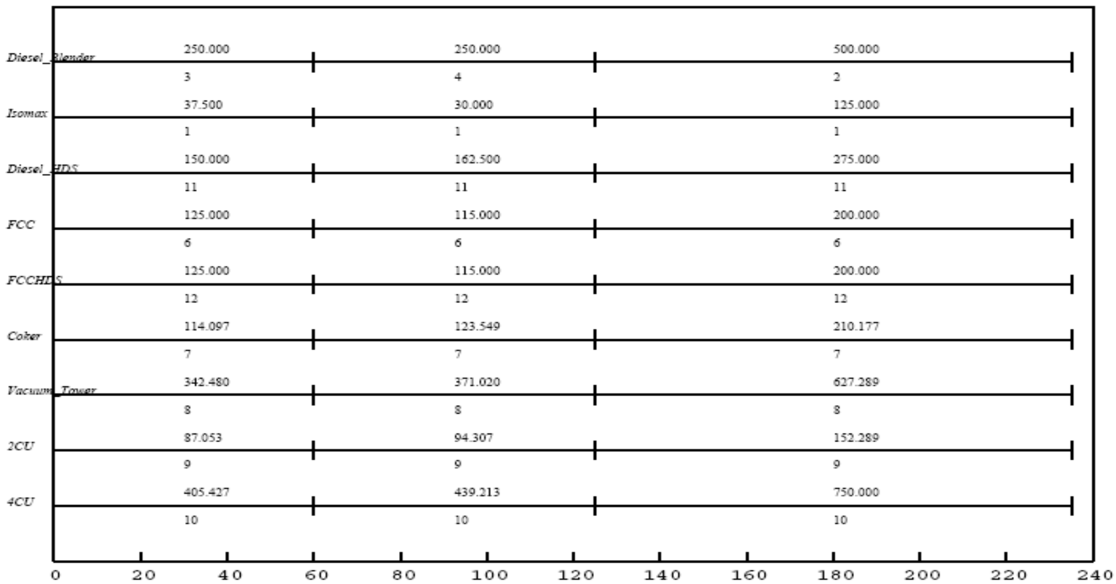

Figure A3-6 Gantt chart of the operation schedule for example 1, Decentralized system, scenario 2

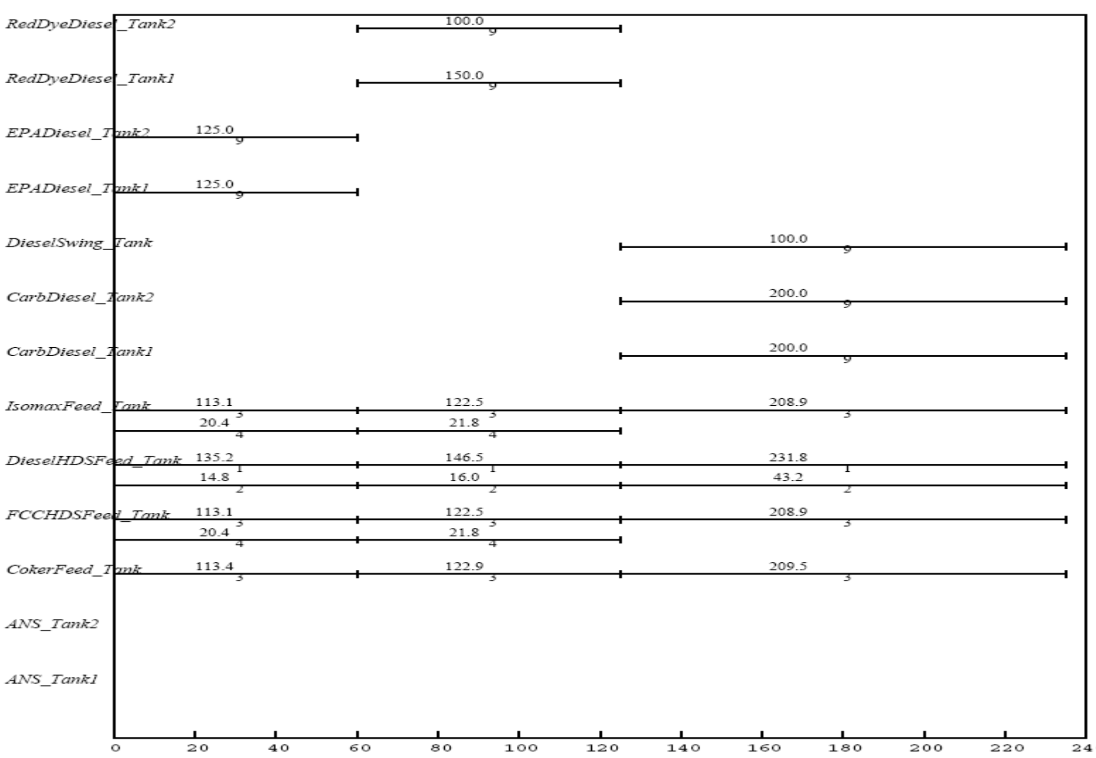

Figure A3-7 Gantt chart of tank loading schedule for example 1, Centralized system, scenario 2

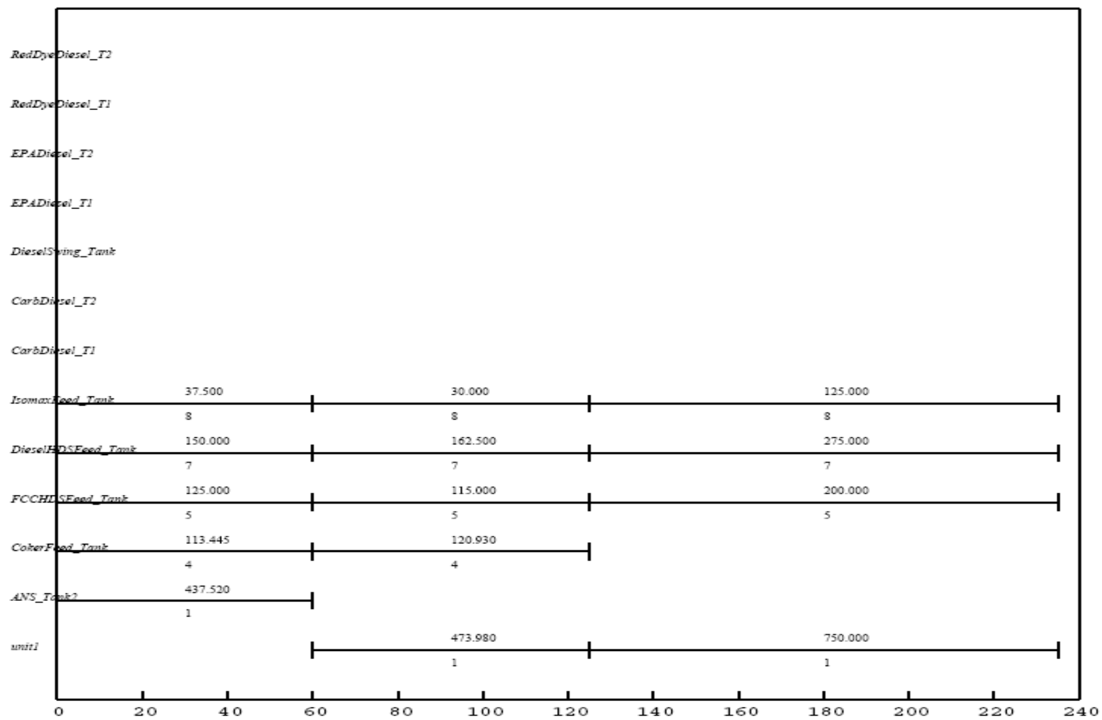

Figure A3-8 Gantt chart of tank unloading schedule for example 1, Centralized system, scenario 2

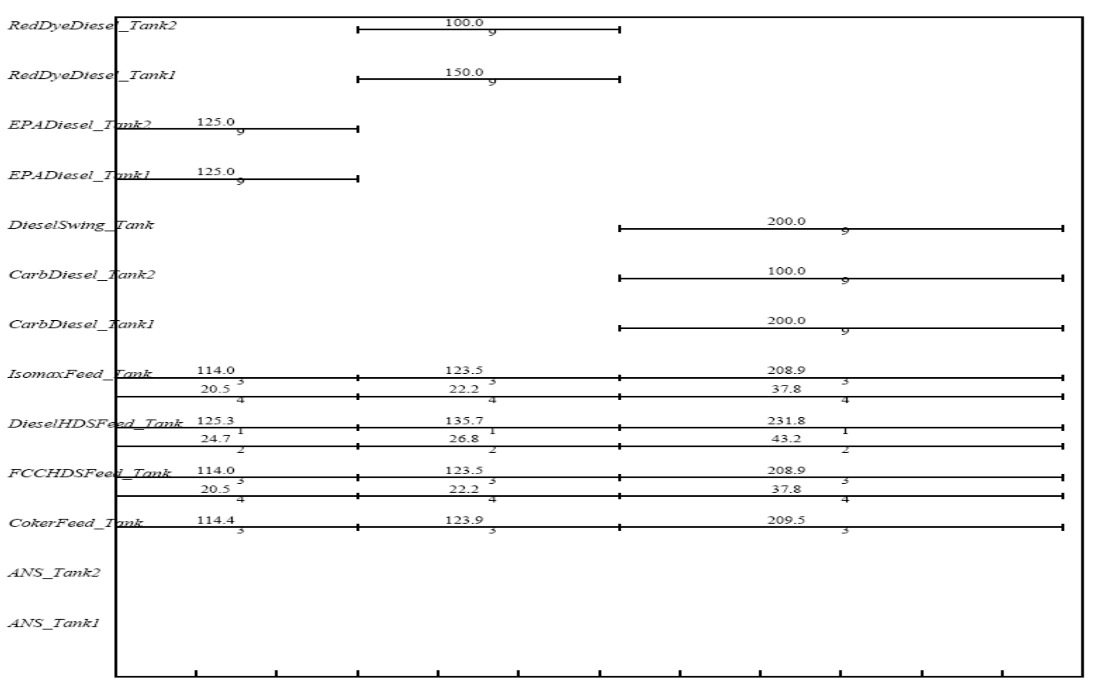

Figure A3-9 Gantt chart of tank loading schedule for example 1, Centralized system, scenario 2

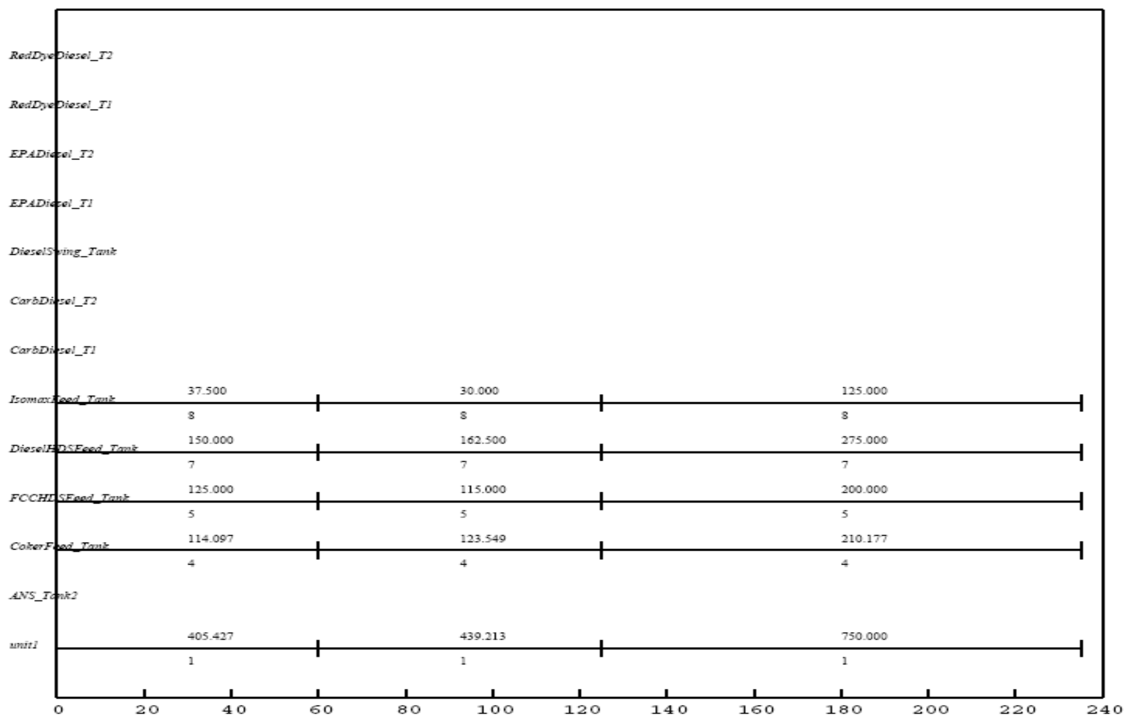

Figure A3-10 Gantt chart of tank unloading schedule for example 1, Centralized system, scenario 2

Appendix Chapter 6

Table A6-1 Production rates in thousand barrels/hour

Unit	**Task**	$\mathbf{R^{min}}$	$\mathbf{R^{max}}$
Diesel Blender	Carb Diesel	0.1	54.167
Diesel Blender	EPA Diesel	0.1	54.167
Diesel Blender	RedDye Diesel	0.1	54.167
Penex	Penex Normal	0.1	0.792
De-C5	De-C5 Normal	0.1	1.042
Alky Unit	Alky Normal	0.1	0.875
Jet Blender	Jet Normal	0.1	3.125
CCR	CCR Normal	0.1	1.75
Isomax	Isomax Normal	0.1	2.042
Naphtha Pre-fractionator	Fractionator Normal	0.1	4.167
FCC	MaxDistillation Mode	0.1	2.708
FCC	MaxGasoline Mode	0.1	2.708
Diesel HDS	Diesel Normal	0.1	2.5
Naphtha HDS	Naphtha Normal	0.1	3.229
FCC HDS	FCCHDS Normal	0.1	3.0
Coker	Coker Normal	0.1	2.75
Vacuum Tower	Vacuum Normal	0.1	5.708

Gas Plant	Gas Normal	0.1	4.167
2Crude Distillation Unit	2CU Normal	0.1	4.167
4Crude Distillation Unit	4CU Normal	0.1	7.292

Table A6-2 Minimum/maximum recipe data

Task	State Produced	$\rho_{s,i}^{p,\min} / \rho_{s,i}^{p,\max}$	State Consumed	$\rho_{s,i}^{c,\min} / \rho_{s,i}^{c,\max}$
Carb Diesel	Carb Diesel	1/1	FCC LCO HDS Diesel HydrocrackedDiesel	0.01/0.25 0.5/0.75 0.05/0.25
EPA Diesel	EPA Diesel	1/1	FCC LCO HDS Diesel HydrocrackedDiesel	0.05/0.3 0.55/0.8 0.01/0.2
RedDye Diesel	Red Dye Diesel	1/1	FCC LCO HDS Diesel HydrocrackedDiesel	0.05/0.3 0.6/0.85 0.01/0.2
Penex Normal	Isomerate	1/1	Light Naphtha Light Ends Butane	0.001/1 0.001/1 0.001/1
De-C5 Normal	Isomax Gasoline Pentane	0.5/1 0.5/1	Light Gasoline	1/1
Alky Normal	Pentane NC4 Alkylate	0.167/0.167 0.167/0.167 0.1/1	Olefins Iso Butanes	0.8/1.125 0.2/0.225

Jet Normal	Jet Fuel	1/1	Isomax Jet Coker Jet SR Jet	0.001/1 0.001/1 0.001/1
CCR Normal	Light Ends Reformate	0.06/0.077 0.7/1	Heavy Naphtha	1/1
Isomax Normal	HydrocrackedDiesel Isomax Jet Heavy Naphtha Light Gasoline Olefins	0.05/1 0.05/1 0.05/1 0.05/1 0.05/1	Light Gasoil	1/1
Fractionator Normal	Heavy Naphtha Light Naphtha	0.001/1 0.001/1	Naphtha HDS	1/1
MaxDistillation Mode	FCC HCO Olefins FCC LCO FCC Gas	0.05/1 0.05/1 0.05/1 0.05/1	FCC HDS	1/1
MaxGasoline Mode	FCC HCO Olefins FCC LCO FCC Gas	0.05/1 0.05/1 0.05/1 0.05/1	FCC HDS	1/1
Diesel Normal	HDS Diesel	1/1	Diesel	1/1
Naphtha Normal	Naphtha HDS	1/1	Naphtha	1/1
FCCHDS Normal	FCC HDS	1/1	Heavy Gasoil	1/1
Coker Normal	Coke Heavy Gasoil	0.28/0.28 0.1/1	Vacuum Resid	1/1

	Light Gasoil Coker Jet Naphtha	0.1/1 0.1/1 0.1/1		
Vacuum Normal	Vacuum Resid Heavy Gasoil Light Gasoil	0.1/1 0.1/1 0.1/1	Resid	1/1
Gas Normal	Iso-Butane Butane Gases	0.001/1 0.001/1 0.001/1	Light Gas	1/1
2CU Normal	Resid Diesel SR Jet Naphtha Light Gas	0.53/0.53 0.21/0.21 0.11/0.11 0.12/0.12 0.03/0.03	SJV crude	1/1
4CU Normal	Resid Diesel SR Jet Naphtha Light Gas	0.47/0.47 0.21/0.21 0.145/0.145 0.14/0.14 0.035/0.035	Crude oil	1/1

Table A6-3 Storage tank capacity data in thousand barrels

Storage Tank	Material Stored	Capacity	Initial Amount
ANS Feed Tank 1	ANS	750	10
ANS Feed Tank 2	ANS	750	10

Coker Feed Tank	Vacuum Resid	500	0
FCC HDS Feed Tank	Heavy Gasoil	500	0
Diesel HDS Feed Tank	Diesel	500	0
Isomax Feed Tank	Light Gasoil	500	0
Carb Diesel Tank 1	Carb Diesel	200	0
Carb Diesel Tank 2	Carb Diesel	200	0
Diesel Swing Tank	EPA Diesel	200	-
Diesel Swing Tank	Carb Diesel	200	-
EPA Diesel Tank 1	EPA Diesel	125	0
EPA Diesel Tank 2	EPA Diesel	125	0
RedDye Diesel Tank1	RedDye Diesel	150	0
RedDye Diesel Tank 2	RedDye Diesel	150	0
Naphtha HDS Feed Tank	Naphtha	500	0
Jet Product Tank 1	Jet Fuel	200	0
Jet Product Tank 2	Jet Fuel	200	0

Table A6-4 Group A products demand order data for the case study

Orders	Orders (Product type, amount(kbbl), delivery window, delivery rate(kbbl/h))									
	Ex1	**Ex2**	**Ex3**	**Ex4**	**Ex5**	**Ex6**	**Ex7**	**Ex8**	**Ex9**	**Ex10**
1	P3 [60,80] [30,50]	P3 [50,70] [20,35]	P2 [100,120] [15,25]	P3 [90,100] [50,75]	P1 [120,125] [40,75]	P3 [60,75] [28.5,46.5]	P1 [25,30] [12,30]	P3 [15,25] [30,50]	P1 [40,50] [30,40]	P2 [20,55] 20,35]
2	P2 [50,70] [60,90]	P4 [50,90] [45,80]	P4 [50,90] [45,80]	P4 [140,150] [60,90]	P2 [100,110] [90,115]	P2 [90,100] [65,89]	P1 [50,65] [50,70]	P2 [40,50] [45,75]	P2 [40,55] [45,58]	P2 [35,50] [40,55]
3	P4 [50,80] [75,100]	P2 [50,75] [60,80]	P3 [50,75] [60,80]	P1 [100,115] [100,120]	P1 [150,175] [135,150]	P4 [140,150] [75,91]	P2 [45,50] [90,105]	P4 [50,75] [75,110]	P1 [45,53] [70,85]	P1 [40,50] [60,75]
4	P1 [50,75] [75,110]	P1 [50,80] [70,90]	P1 [50,80] [70,90]	P4 [100,120] [125,150]	P2 [90,120] [168,185]	P1 [90,115] [102,116]	P4 [90,110] [98,128]	P2 [20,30] [75,95]	P3 [40,44] [92,105]	P3 [15,60] [80,100]
5				P2 [110,125] [130,150]	P3 [105,125] [200,220]	P4 [100,150] [105,130]	P3 [50,65] [125,141]	P3 [10,20] [90,125]	P2 [30,60] [110,125]	P2 [30,50] [105,115]

6				P1 [50,60] [160,175]	P4 [120,150] [50,75]	P1 [100,115] [125,153]	P2 [35,40] [160,180]	P1 [15,30] [100,115]	P3 [35,50] [120,140]	P3 [40,45] [110,125]
7					P4 [100,120] [120,146]	P4 [100,165] [155,170]	P4 [100,115] [160,193]	P1 [55,60] [115,145]	P1 [35,40] [145,160]	P1 [30,65] [130,145]
8					P4 [160,170] [175,205]	P2 [75,100] [179,203]	P3 [45,65] [175,205]	P4 [75,100] [115,145]	P3 [30,75] [175,190]	P1 [25,50] [150,165]
9						P4 [120,125] [195,210]	P4 [85,100] [200,219]	P4 [80,100] [150,175]	P2 [40,40] [195,210]	P2 [30,45] [170,185]
10						P3 [75,100] [200,215]	P1 [70,90] [210,227]	P2 [50,65] [155,170]	P1 [55,75] [205,220]	P3 [25,55] [170,190]
11							P2 [50,65] [220,240]	P3 [10,15] [160,180]	P2 [40,45] [205,225]	P1 [20,55] [200,220]

12							P4 [120,140] [223,240]	P3 [35,40] [180,200]	P1 [50,60] [220,240]	P1 [25,45] [215,235]
13							P3 [60,80] [228,240]	P2 [10,15] [200,220]	P4 [50,75] [50,75]	P3 [30,40] [220,240]
14								P4 [100,115] [200,230]	P4 [75,115] [100,120]	P4 [40,60] [30,65]
15								P1 [30,45] [210,235]	P4 [60,110] [125,150]	P4 [55,95] [55,75]
16								P4 [75,100] [220,240]	P4 [55,120] [140,165]	P4 [50,65] [95,110]
17									P4 [85,125] [175,200]	P4 [30,105] [115,125]

18									P4 [100,125] [205,225]	P4 [60,125] [130,155]
19									P4 [78,100] [220,240]	P4 [75,105] [160,180]
20										P4 [85,115] [180,205]
21										P4 [100,120] [200,225]
22										P4 [110,130] [215,240]

Table A6-5 Group B products demands data for the case study

Ex.	Initial holdup in swing	Group B products demand (kbbl)

	tank (product, kbbl)	P5	P6	P7	P8	P9	P10	P11	P12	P13
1	-	5	5	10	10	10	5	4	5	5
2	P1 - 10	10	5	10	15	10	10	5	15	10
3	P1 - 10	5	10	15	10	5	5	10	25	5
4	-	10	25	25	20	5	10	10	15	0
5	-	15	30	25	25	15	13	10	50	10
6	-	13	25	30	10	20	15	17	75	5
7	P1 - 10	10	20	30	25	30	15	10	60	15
8	-	20	15	25	20	25	10	15	95	20
9	P1 - 10	15	35	35	20	20	15	20	70	10
10	-	10	25	35	25	15	16	10	80	10

Appendix Chapter 7

Valid Inequalities

There are several constraints, redundant in the model but including them improves computational time. We propose following inequalities for decomposed models:

$$x_{j,n,t} \leq \sum_{t'<t, n'\geq t'} x_{j,n',t'}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, n \geq t, t > 1, n > 1 \tag{1a}$$

$$x_{j,n,t'} \leq \sum_{n'\geq t} x_{j,n',t}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N, t \in T, t' \in T, t' > t, n \geq t' \tag{1b}$$

$$xk_{s,n,t} \leq \sum_{t'<t, n'\geq t'} xk_{s,n',t'}, \quad \forall s \in S_k, n \in N, t \in T, n \geq t, t > 1, n > 1 \tag{1c}$$

$$xk_{s,n,t'} \leq \sum_{n'\geq t} xk_{s,n',t}, \quad \forall s \in S_k, n \in N, t \in T, t' \in T, t' > t, n \geq t' \tag{1d}$$

Eqs. (1a)-(1d) restates that transfer event-point $t < t'$ should be assigned an active event-point n before t' and $n < n'$ should be assigned a transfer event-point before n'. These inequalities are added to both (LBSP) and (LPSP).

Relaxed Blend Scheduling Problem:

$$y_{s,k,n} \leq \sum_{j \in J_k^{pk}} in_{s,j,k,n} + yo_{s,k} + \sum_{s' \in S_k, \varphi_s < \varphi_{s'}} yo_{s',k}, \quad \forall s \in \left(S_A \cap S_{k \in K^m}\right) \cup S_{bc}, k \in K_s, n = 1 \tag{2a}$$

$$y_{s,k,n} \leq \sum_{j \in J_k^{pk}} in_{s,j,k,n} + y_{s,k,n-1} + \sum_{s' \in S_k, \varphi_s < \varphi_{s'}} y_{s',k,n-1}, \quad \forall s \in \left(S_A \cap S_{k \in K^m}\right) \cup S_{bc}, k \in K_s, n \in N, n > 1 \tag{2b}$$

Inequalities (2a)-(2b) states that if material is stored in tank at event point n then it either be hold up from previous event-point or flow of the material in to the tank at the same event-point. Above constraint is only included for blend components and finished blend products that can be stored in multipurpose (swing) product tanks (K^m).

$$\sum_{j \in J^{BU} \cap J^{ds}, i \in I_j, n' > 1} wv_{i,j,n'} \leq \left(|T| - 1\right) \sum_{j \in J^{ds}} \left(\sum_{j \in J^{ds}, i \in I_j} wv_{i,j,n} \right), \quad \forall bn \in BN, n = 1, \exists j \in J^{ds} \tag{3a}$$

$$\sum_{j \in J^{BU} / J^{ds}, i \in I_j, n' > 1} wv_{i,j,n'} \leq \left(|N| - 1\right) \sum_{j \in J^{BU}, j \notin J^{ds}} \left(\sum_{j \notin J^{ds}, i \in I_j} wv_{i,j,n} \right), \quad \forall bn \in BN, n = 1, \exists j \notin J^{ds} \tag{3b}$$

$$\sum_{i\in I_j,n} wv_{i,j,n} \le |T|, \quad \forall j \in J^{BU} \cap J^{ds}, \sum_s \overline{x}^P_{s,j,t} < |T| \tag{3c}$$

$$\sum_n in_{s,j,k,n} \le |T|, \quad \forall s \in S_{bc}, k \in K_s, j \in J_{dmy}, \sum_s \overline{xk}^P_{s,t} < |T| \tag{3d}$$

We restrict a feasible solution of decomposed scheduling problem by always requiring the production to take place during first event-point using Eqs. (3a) and (3b) and these inequalities do not eliminate optimal solution. For blend unit that receive at least one blend components straight from production unit operations without any storage, the blend unit would be operational only when it is receiving component from PN. Thus, the total run-modes for the blend unit are bounded by total transfer event-points as stated by Eq. (3c). Similarly, inequality (3d) stresses the flow of component into a tank also bounded.

$$\sum_{n\ge t} x^B_{j,n,t} \le 1, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T \tag{4a}$$

$$\sum_{n\ge t} xk^B_{s,n,t} \le 1, \quad \forall s \in S_k, t \in T \tag{4b}$$

$$\sum_{n\ge t} x^B_{j,n,t} \le \sum_{i\in I_j, n\ge t} wv_{i,j,n}, \quad \forall j \in J^{BU} \cap J^{ds}, t \in T \tag{4c}$$

$$\sum_{t\le n} x^B_{j,n,t} = \sum_{i\in I_j} wv_{i,j,n}, \quad \forall j \in J^{BU} \cap J^{ds}, n \in N \tag{4d}$$

$$\sum_{n\ge t} xk^B_{s,n,t} \le \sum_{k\in K_s, j\in J_{dmy}, n\ge t} in_{s,j.k,n}, \quad \forall s \in S_k, t \in T \tag{4e}$$

$$\sum_{t\le n} xk^B_{s,n,t} \le \sum_{k\in K_s, j\in J_{dmy}} in_{s,j.k,n}, \quad \forall s \in S_k, n \in N \tag{4f}$$

$$\sum_{k\in K_s} \sum_{t\le n} xk^B_{s,n,t} \ge \sum_{k\in K_s, j\in J_{dmy}} in_{s,j.k,n}, \quad \forall s \in S_k, n \in N \tag{4g}$$

If flow is happening between PN and BN at event-point n then it must corresponds to at most one transfer event-point t as restated by Eqs. (4a) and (4b). If there is flow at event point t then it must correspond to some event-point n and this is reinforced by Eqs. (4c)-(4g).

Relaxed Production Unit Scheduling Problem:

$$\sum_{s\in S_{bc}^{bn},i\in I_s^p,j'\in J_s^c\cap J^{ds},j\in J_i,n'>1} wv_{i,j,n'} \le \left(|T|-1\right) \sum_{s\in S_{bc}^{bn},j\in J_s^p,j'\in J_s^c\cap J^{ds}} \left(\sum_{s\in S_{bc}^{bn},i\in I_s^p,j'\in J_s^c,j\in J_i} wv_{i,j,n} \right), \quad \forall bn\in BN, n=1, \exists j\in J^{ds} \tag{5}$$

Valid inequality (5) limits the feasible solution space by requiring production to take place during first event-point ($n=1$) for production units that supply components directly to a blend unit.

$$\sum_{j'\in J^{BU}\cap J_s^c} a_{j,j',n} \le \sum_{j'\in J^{BU}\cap J_s^c} \sum_{i\in I_s^p} wv_{i,j,n}, \quad \forall s\in S_{bc}/S_k, j\in J_s^p, n\in N \tag{6a}$$

$$\sum_{j'\in J^{BU}\cap J_s^c} a_{j,j',n} \ge \sum_{i\in I_s^p} wv_{i,j,n}, \quad \forall s\in S_{bc}/S_k, j\in J_s^p, n\in N \tag{6b}$$

If a blend component that has no storage option is being produced by a production unit, then the production unit will be sending this blend component to suitable blend unit and this material balance relation is restated in valid inequalities (6a) and (6b).

$$\sum_{t\le n} x_{j,n,t}^P \le \sum_{i\in I_s^p,j'\in J_i} wv_{i,j',n}, \quad \forall s\in S_{bc}/S_k, j\in J^{BU}, n\in N \tag{7a}$$

$$\sum_{j'\in J_s^p,t\le n} x_{j,n,t}^P \ge \sum_{i\in I_s^p,j'\in J_i} wv_{i,j',n}, \quad \forall s\in S_{bc}/S_k, j\in J^{BU}/J^h, n\in N \tag{7b}$$

$$\sum_{n\in N,t\le n} x_{j,n,t}^P = \sum_{i\in I_s^p,j'\in J_i,n} wv_{i,j',n}, \quad \forall s\in S_{bc}/S_k, j\in J^{BU}/J^h \tag{7c}$$

$$\sum_{j\in J_{bn}^{BU},j'\in J_s^p,t\le n} x_{j,n,t}^P \ge \sum_{i\in I_s^p,j'\in J_i} wv_{i,j',n}, \quad \forall s\in S_{bc}/S_k, bn\in BN, n\in N, \exists j\in J_{bn}^{BU}\cap J^h \tag{7d}$$

$$\sum_{t\le n} x_{j,n,t}^P \le \sum_{j'\in J_s^p} a_{j',j,n}, \quad \forall s\in S_{bc}/S_k, j\in J^{BU}, n\in N \tag{7e}$$

$$\sum_{t\le n} x_{j,n,t}^P \ge a_{j',j,n}, \quad \forall s\in S_{bc}/S_k, j\in J^{BU}, j'\in J_s^p, n\in N \tag{7f}$$

$$\sum_{t\le n} xk_{s,n,t}^P \le \sum_{i\in I_s^p,j\in J_i} wv_{i,j,n}, \quad \forall s\in S_{bc}\cap S_k, n\in N \tag{7g}$$

$$\sum_{j\in J_s^p,t\le n} xk_{s,n,t}^P \ge \sum_{i\in I_s^p,j\in J_i} wv_{i,j,n}, \quad \forall s\in S_{bc}\cap S_k, n\in N \tag{7h}$$

$$\sum_{n\in N,t\le n} xk_{s,n,t}^P = \sum_{i\in I_s^p,j\in J_i,n} wv_{i,j,n}, \quad \forall s\in S_{bc}\cap S_k \tag{7i}$$

Valid inequalities (7a)-(7f) restates the relationship between production units that supply components directly to blend units and transfer event-point t and production event-point n. Eqs.

(7g)-(7i) emphasis relationship between production units that supply components to BN for storage at event-point n and to transfer event-point t. These inequalities speed up convergence to optimal solution by decreasing the number of iterations and nodes in branch and bound tree in solver CPLEX.

When the optimization model is written in GAMS environment, where the variables and equations are declared have significant impact on the solution time (GAMS TUTIORIAL). In our work, we have added all the inequalities presented in this section to our relaxed models (LBSP) and (LPSP). However, depending upon how the model is written in GAMS, one or more of these inequalities might not be necessary.

Table A7-1 Production rates (thousand barrels/hour) for a second diesel blend unit

Diesel Blender2	Carb Diesel	0.1	34.613
Diesel Blender2	EPA Diesel	0.1	34.613
Diesel Blender2	RedDye Diesel	0.1	34.613

Table A7-2 Storage tank capacity data for blend component tanks

Storage Tank	Material Stored	Capacity	Initial Amount
FCC LCO Tank	FCC LCO	300	0
HDS Diesel Tank	HDS Diesel	500	0
HydrocrackedDiesel Tank	HydrocrackedDiesel	350	0
Isomax Jet Tank	Isomax Jet	500	0
Coker Jet Tank	Coker Jet	500	0
SR Jet Tank	SR Jet	500	0

Table A7-3 Group B products demands data for the case study

Ex.	Initial holdup in swing tank (product, kbbl)	Group B products demand (kbbl)								
		P5	P6	P7	P8	P9	P10	P11	P12	P13
1	-	5	5	10	10	10	5	4	5	5
2	-	2	14	30	2	4	3	6	4	30
3	P1 - 10	10	5	10	15	10	10	5	15	10
4	-	10	25	25	20	5	10	10	15	0
5	-	15	30	25	25	15	13	10	50	10
6	P1-10	12	25	25	15	15	12	10	45	10
7	P1 - 10	10	20	30	25	30	15	10	60	15

Table A7-4 Group A products demand order data for the case study

Orders	Orders (Product type, amount(kbbl), delivery window, delivery rate(kbbl/h))						
	Ex1	**Ex2**	**Ex3**	**Ex4**	**Ex5**	**Ex6**	**Ex7**
1	P3 [60,80] [30,50]	P3 [60,80] [30,50]	P3 [50,70] [20,35]	P3 [90,100] [50,75]	P1 [120,125] [40,75]	P3 [100,115] [40,70]	P1 [25,30] [12,30]
2	P2 [50,70] [60,90]	P2 [50,70] [60,90]	P4 [50,90] [45,80]	P4 [140,150] [60,90]	P2 [100,110] [90,115]	P2 [100,110] [70,100]	P1 [50,65] [50,70]
3	P4 [50,80] [75,100]	P4 [50,80] [75,100]	P2 [50,75] [60,80]	P1 [100,115] [100,120]	P1 [150,175] [135,150]	P1 [150,175] [115,145]	P2 [45,50] [90,105]
4	P1 [50,75] [75,110]	P1 [50,75] [75,110]	P1 [50,80] [70,90]	P4 [100,120] [125,150]	P2 [90,120] [168,185]	P2 [90,120] [180,215]	P4 [90,110] [98,128]
5				P2 [110,125] [130,150]	P3 [105,125] [200,220]	P1 [100,115] [200,225]	P3 [50,65] [125,141]

6				P1 [50,60] [160,175]	P4 [120,150] [50,75]	P4 [110,130] [60,98]	P2 [35,40] [160,180]
7					P4 [100,120] [120,146]	P4 [120,140] [124,155]	P4 [100,115] [160,193]
8					P4 [160,170] [175,205]	P4 [160,170] [195,235]	P3 [45,65] [175,205]
9							P4 [85,100] [200,219]
10							P1 [70,90] [210,227]
11							P2 [50,65] [220,240]

12							P4 [120,140] [223,240]
13							P3 [60,80] [228,240]

Bibliography

ACC. (2014). Chemistry industry fact. In. http://www.americanchemistry.com/chemistry-industry-facts: American Chemistry Council.

Adhya, N., Tawarmalani, M., & Sahinidis, N. V. (1999). A Lagrangian Approach to the Pooling Problem. Industrial & Engineering Chemistry Research, 38, 1956-1972.

Amaldi, E., Bruglieri, M., & Casale, G. (2008). A two-phase relaxation-based heuristic for the maximum feasible subsystem problem. Computers & Operations Research, 35, 1465-1482.

Andreani, R., Birgin, E. G., Mart´ınez, J. M., & Schuverdt, M. L. (2008). On Augmented Lagrangian methods with general lower-level constraints. SIAM Journal on Optimization, 18, 1286-1309.

Arief, A., Rajagopalan, S., & Karimi, I. A. (2007). Heuristic rescheduling of crude oil operations to manage abnormal supply chain events. AIChE Journal, 53, 397-422.

Audet, C., Brimberg, J., Hansen, P., Digabel, S. L., & Mladenovic, N. (2004). Pooling Problem: Alternate Formulations and Solution Methods. Management Science, 50, 761-776.

Bassett, M. H., Pekny, J., & Reklaitis, G. (1996). Decomposition Techniques for the Solution of Large-Scale Scheduling Problems. American Institute of Chemical Engineers Journal, 42, 3373-3387.

Benders, J. F. (1962). Partitioning Procedures for Solving Mixed-Variables Programming Problems. Numerische Mathematic, 4, 238-252.

Bertsekas, D. P. (1995). Nonlinear Programming (2nd ed. ed.). Belmont, Massachusetts: Athena Scientific.

Bertsekas, D. P., & Tsitsiklis, J. N. (1989). Parallel and Distributed Computation. Englewood Cliffs, New Jersey: Prentice-Hall.

Bertsimas, D., & Sim, M. (2003). Robust Discrete optimization and Network Flows. Math. Prog., 98, 49-71.

Burkard, R. E., Fortuna, T., & Hurkens, C. A. J. (2002). Makespan minimization for chemical batch processes using non-uniform time grids. Computers & Chemical Engineering, 26, 1321-1332.

Calfa, B. A., Agarwal, A., Grossmann, I. E., & Wassick, J. M. (2013). Hybrid Bilevel-Lagrangean Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants. Industrial & Engineering Chemistry Research, 52, 2152-2167.

Castillo, P. A. C., & Mahalec, V. (2014a). Inventory pinch based, multiscale models for integrated planning and scheduling-part I: Gasoline blend planning. AIChE Journal, 60, 2158-2178.

Castillo, P. A. C., & Mahalec, V. (2014b). Inventory pinch based, multiscale models for integrated planning and scheduling-part II: Gasoline blend scheduling. AIChE Journal, 60, 2475-2497.

Castro, P. M., & Grossmann, I. E. (2014). Global Optimal Scheduling of Crude Oil Blending Operations with RTN Continuous-time and Multiparametric Disaggregation. Industrial & Engineering Chemistry Research, 53, 15127-15145.

Castro, P. M., Harjunkoski, I., & Grossmann, I. E. (2009). Optimal Short-Term Scheduling of Large-Scale Multistage Batch Plants. Industrial & Engineering Chemistry Research.

Castro, P. M., Harjunkoski, I., & Grossmann, I. E. (2011). Greedy algorithm for scheduling batch plants with sequence-dependent changeovers. AIChE Journal, 57, 373-387.

Chen, J. M., & Chen, T. H. (2005). The Multi-item Replenishment Problem in a Two-Echelon Supply Chain: The Effect of Centralization versus Decentralization. Computers & Operations Research, 32, 3191-3207.

Cornillier, F., Boctor, F. F., Laporte, G., & Renaud, J. (2008). A heuristic for the multi-periodpetrol station replenishment problem. European Journal of Operational Research, 191, 295-305.

Cornuejols, G., Fisher, M., & Nemhauser, G. L. (1977). On the Uncapacitated Location Problem*. In E. L. J. B. H. K. P.L. Hammer & G. L. Nemhauser (Eds.), Annals of Discrete Mathematics (Vol. Volume 1, pp. 163-177): Elsevier.

Cuiwen, C., Xingsheng, G., & Zhong, X. (2013). A data-driven rolling-horizon online scheduling model for diesel production of a real-world refinery. AIChE Journal, 59, 1160-1174.

Dimitriadis, A. D., Shah, N., & Pantelides, C. C. (1997). RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. Computers & Chemical Engineering, 21, S1061-S1066.

Fisher, M. L. (1973). Optimal Solution of Scheduling Problems Using Lagrange Multipliers: Part I. Operations Research, 21, 1114-1127.

Fisher, M. L. (1981). The Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science, 27, 1-18.

Fisher, M. L. (1985). An Applications Oriented Guide to Lagrangian Relaxation. Interfaces, 15, 10-21.

Floudas, C. A., & Lin, X. (2004). Continuous-Time Versus Discrete-Time Approaches For Scheduling of Chemical Processes: a Review. Computers & Chemical Engineering, 28, 2109-2129.

Fortin, M., & Glowinski, R. (1983). Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-Value Problems: North-Holland.

Gaivoronski, A. (1988). Stochastic quasigradient methods and their implementation. In Numerical techniques for stochastic optimization (Vol. 10, pp. 313-351): Springer.

Gary, J. H., Handwerk, G. E., & Kaiser, M. J. (2007). Petroleum Refining: Technology and Economics: CRC Press; 5 edition.

Ghaddar, B., Naoum-Sawaya, J., Kishimoto, A., Taheri, N., & Eck, B. (2014). Innovative Applications of O.R.: A Lagrangian decomposition approach for the pump scheduling problem in water networks. European Journal of Operational Research.

Glismann, K., & Gruhn, G. (2001). Short-term scheduling and recipe optimization of blending processes. Computers & Chemical Engineering, 25, 627-634.

Gnoni, M. G., Iavagnilio, R., Mossa, G., Mummolo, G., & Di Leva, A. (2003). Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry. International Journal of Production Economics, 85, 251-262.

Gounaris, C. E., Misener, R., & Floudas, C. A. (2009). Computational Comparison of Piecewise Linear Relaxations for Pooling Problems. Industrial & Engineering Chemistry Research, 48, 5742-5766.

Graves, S. C. (1982). Using Lagrangean Techniques to Solve Hierarchical Production Planning Problems. Management Science, 28, 260-275.

Grossmann, I. (2005). Enterprise-wide Optimization: A New Frontier in Process Systems Engineering. AIChE Journal, 51, 1846-1857.

Grossmann, I. E. (2014). Challenges in the application of mathematical programming in the enterprise-wide optimization of process industries. Theoretical Foundations of Chemical Engineering, 48, 555-573.

Gupta, A., & Maranas, C. D. (1999). A Hierarchical Lagrangean Relaxation Procedure for Solving Midterm Planning Problems. Industrial & Engineering Chemistry Research, 38, 1937-1947.

Gupta, A., Maranas, C. D., Rafiqul, G., & Sten Bay, J. (2001). Multiperiod planning of multisite supply chains under demand uncertainty. In COMPUTER AIDED CHEMICAL ENGINEERING (Vol. Volume 9, pp. 871-876): Elsevier.

Harjunkoski, I., & Grossmann, I. E. (2001). A Decomposition Approach For the Scheduling of a Steel Plant Production. Computers and Chemical Engineering, 25, 1647-1660.

Harjunkoski, I., Maravelias, C. T., Bongers, P., Castro, P. M., Engell, S., Grossmann, I. E., Hooker, J., Méndez, C., Sand, G., & Wassick, J. (2014). Scope for industrial applications of production scheduling models and solution methods. Computers & Chemical Engineering, 62, 161-193.

He, Y., & Hui, C.-W. (2007). Genetic algorithm for large-size multi-stage batch plant scheduling. Chemical Engineering Science, 62, 1504-1523.

Held, M., & Karp, R. (1971). The traveling-salesman problem and minimum spanning trees: Part II. Mathematical Programming, 1, 6-25.

Held, M., & Karp, R. M. (1970). The Traveling-Salesman Problem and Minimum Spanning Trees. Operations Research, 18, 1138-1162.

Holmberg, K., & Hellstrand, J. (1998). Solving the Uncapacitated Network Design Problem by a Lagrangean Heuristic and Branch-and-Bound. Operations Research, 46, 247-259.

Holmberg, K., & Yuan, D. (2000). A Lagrangian Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem. Operations Research, 48, 461-481.

Ierapetritou, M. G., & Floudas, C. (1998). Effective Continuous-time Formulation for Short-Term Scheduling: 1 Multipurpose Batch Processes Industrial & Engineering Chemical Research, 37, 4341-4359.

Ierapetritou, M. G., & Floudas, C. A. (1998). Effective Continuous-Time Formulation for Short-Term Scheduling. 2. Continuous and Semicontinuous Processes. Industrial & Engineering Chemistry Research, 37, 4360-4374.

Ierapetritou, M. G., & Floudas, C. A. (1998). Effective Continuous-Time Formulation for Short-Term Scheduling: 2 Multipurpose/Multiproduct Continuous Processes. Industrial & Engineering Chemical Research, 37, 4360-4374.

Ierapetritou, M. G., Hene, T. S., & Floudas, C. A. (1999). Effective Continuous-Time Formulation for Short -Term Scheduling: 3. Multiple Intermediate Due Dates. Industrial & Engineering Chemical Research, 38, 3446-3461.

Jackson, J. R., & Grossmann, I. E. (2003). Temporal Decomposition Scheme for Nonlinear Multisite Production Planning and Distribution Models. Industrial & Engineering Chemistry Research, 42, 3045-3055.

Janak, S. L., Floudas, C. A., Kallrath, J., & Vormbrock, N. (2006). Production Scheduling of a Large-Scale Industrial Batch Plant. I. Short-Term and Medium-Term Scheduling. Industrial & Engineering Chemistry Research, 45, 8234-8252.

Janak, S. L., Lin, X., & Floudas, C. A. (2004). Enhanced Continuous-Time Unit-Specific Event-Based Formulation for Short-Term Scheduling of Multipurpose Batch Processes: Resource Constraints and Mixed Storage Policies. Industrial & Engineering Chemistry Research, 43, 2516-2533.

Jia, Z., & Ierapetritou, M. (2003). Mixed-Integer Linear Programming Model for Gasoline Blending and Distribution Scheduling. Industrial & Engineering Chemistry Research, 42, 825-835.

Jia, Z., & Ierapetritou, M. (2004). Efficient short-term scheduling of refinery operations based on a continuous time formulation. Computers & Chemical Engineering, 28, 1001-1019.

Jia, Z., Ierapetritou, M., & Kelly, J. D. (2003). Refinery Short-Term Scheduling Using Continuous Time Formulation: Crude-Oil Operations. Industrial & Engineering Chemistry Research, 42, 3085-3097.

Joly, M., Moro, L. F. L., & Pinto, J. M. (2002). Planning and scheduling for petroleum refineries using mathematical programming. Brazilian Journal of Chemical Engineering, 19, 207-228.

Kallrath, J. (2002a). Combined strategic and operational planning – an MILP success story in chemical industry. OR Spectrum, 24, 315-341.

Kallrath, J. (2002b). Planning and scheduling in the process industry. OR Spectrum, 24, 219-250.
Karuppiah, R., Furman, K. C., & Grossmann, I. E. (2008). Global optimization for scheduling refinery crude oil operations. Computers & Chemical Engineering, 32, 2745-2766.
Karuppiah, R., Martín, M., & Grossmann, I. E. (2010). A simple heuristic for reducing the number of scenarios in two-stage stochastic programming. Computers & Chemical Engineering, 34, 1246-1255.
Kelly, J. D. (2003a). Next-generation refinery scheduling technology. NPRA Plant Automation and Decision Support Conference, San Antonio, TX, Presentation.
Kelly, J. D. (2003b). Smooth-and-dive accelerator, a pre-MILP primal heuristic applied to scheduling. Computer & Chemical Engineering 27, 873-832.
Kelly, J. D. (2006). Logistics: the missing link in blend scheduling optimization. Hydrocarbon Processing, 45-51.
Kelly, J. D., & Mann, J. L. (2003a). Crude-oil blend scheduling optimization: An application with multi-million dollar benefits: Part I. Hydrocarbon Processing, 47-53.
Kelly, J. D., & Mann, J. L. (2003b). Crude-oil blend scheduling optimization: An application with multi-million dollar benefits: Part II. Hydrocarbon Processing, 72-79.
Kelly, J. D., & Zyngier, D. (2008). Hierarchical Decomposition Heuristic for Scheduling: Coordinated Reasoning for Decentralized And Distributed Decision-Making Problems. Computers and Chemical Engineering, 32, 2684-2705.
Khor, C. S., Chachuat, B., & Shah, N. (2012). A superstructure optimization approach for water network synthesis with membrane separation-based regenerators. Computers & Chemical Engineering, 42, 48-63.
Kolodziej, S. P., Grossmann, I. E., Furman, K. C., & Sawaya, N. W. (2013). A discretization-based approach for the optimization of the multiperiod blend scheduling problem. Computers & Chemical Engineering, 53, 122-142.
Kondili, E. (1987). Optimal Scheduling of Batch Processes. Imperial College London, U.K.
Kondili, E., Pantelides, C. C., & Sargent, R. W. H. (1993). A General Algorithm for Short-Term Scheduling of Batch Operations--I. MILP Formulation. Computers & Chemical Engineering, 17, 211-227.
Kreipl, S., & Pinedo, M. (2004). Planning and Scheduling in Supply Chains: An Overview of Issues in Practice. Production and Operations Management, 13, 77-92.
Lee, H. M., Pinto, J. M., Grossmann, I. E., & Park, S. (1996). Mixed-integer linear programming model for refinery short-term scheduling of crude oil unloading with inventory management. Industrial & Engineering Chemistry Research, 35, 1630-1641.
Li, J., & Karimi, I. A. (2011). Scheduling Gasoline Blending Operations from Recipe Determination to Shipping Using Unit Slots. Industrial & Engineering Chemistry Research, 50, 9156-9174.
Li, J., Karimi, I. A., & Srinivasan, R. (2010). Recipe determination and scheduling of gasoline blending operations. AIChE Journal, 56, 441-465.
Li, Y., Lu, Z., & Michalek, J. J. (2008). Diagonal Quadratic Approximation for Parallelization of Analytical Target Cascading. Journal of Mechanical Design, 130, 051402-051411.
Li, Z., & Ierapetritou, M. G. (2010a). Production planning and scheduling integration through augmented Lagrangian optimization. Computers & Chemical Engineering, 34, 996-1006.
Li, Z., & Ierapetritou, M. G. (2010b). Rolling horizon based planning and scheduling integration with production capacity consideration. Chemical Engineering Science, 65, 5887-5900.
Lin, J., & Chen, Y.-Y. (2007). A multi-site supply network planning problem considering variable time buckets– A TFT-LCD industry case. The International Journal of Advanced Manufacturing Technology, 33, 1031-1044.
Luh, P. B., & Hoitomt, D. J. (1993). Scheduling of manufacturing systems using the Lagrangian relaxation technique. Automatic Control, IEEE Transactions on, 38, 1066-1079.

Luo, C., & Rong, G. Hierarchical Approach for Short-Term Scheduling in Refineries. Industrial & Engineering Chemistry Research, 49, 4474-4474.

Luo, C., & Rong, G. (2007). Hierarchical Approach for Short-Term Scheduling in Refineries. Industrial & Engineering Chemistry Research, 46, 3656-3668.

Maravelias, C. T., & Grossmann, I. E. (2003). New General Continuous-Time State鈭扵ask Network Formulation for Short-Term Scheduling of Multipurpose Batch Plants. Industrial & Engineering Chemistry Research, 42, 3056-3074.

Maravelias, C. T., & Sung, C. (2009). Integration of production planning and scheduling: Overview, challenges and opportunities. Computers & Chemical Engineering, 33, 1919-1930.

Mendez, C. A., Grossmann, I. E., Harjunkoski, I., & Kabore, P. (2006). A simultaneous optimization approach for off-line blending and scheduling of oil-refinery operations. Computers & Chemical Engineering, 30, 614-634.

Minoux, M. (1986). Mathematical Programming Theory and Algorithms.

Moniz, S., Barbosa-Póvoa, A. P., De Sousa, J. P., & Duarte, P. (2014). Solution methodology for scheduling problems in batch plants. Industrial and Engineering Chemistry Research, 53, 19265-19281.

Moro, L. F. L., Zanin, A. C., & Pinto, J. M. (1998). A planning model for refinery diesel production. Computers & Chemical Engineering, 22, S1039-S1042.

Mouret, S., Grossmann, I. E., & Pestiaux, P. (2010). Time Representations and Mathematical Models for Process Scheduling Problems. Computers & Chemical Engineering, 35, 1038-1063.

Mouret, S., Grossmann, I. E., & Pestiaux, P. (2011). A new Lagrangian decomposition approach applied to the integration of refinery planning and crude-oil scheduling. Computers & Chemical Engineering, 35, 2750-2766.

Neiro, S. M. S., Murata, V. V., & Pinto, J. M. (2014). Hybrid Time Formulation for Diesel Blending and Distribution Scheduling. Industrial & Engineering Chemistry Research, 53, 17124-17134.

Nishi, T., Shinozaki, R., & Konishi, M. (2008). An augmented Lagrangian approach for distributed supply chain planning for multiple companies. Ieee Transactions on Automation Science and Engineering, 5, 259-274.

. The Outlook for Energy: A View to 2040. In. (2013): ExxonMobil.

Pantelides, C. C. (1994). Unified Frameworks for Optimal Process Planning and Scheduling. In D. W. T. Rippin & J. Hale (Eds.), Proceedings of the Second Conference on Foundations of Computer-Aided Process Operations (pp. 253): CACHE Publications.

Papageorgiou, L. G. (2009). Supply chain optimisation for the process industries: Advances and opportunities. Computers & Chemical Engineering, 33, 1931-1938.

Pinto, J. M., Joly, M., & Moro, L. F. L. (2000). Planning and scheduling models for refinery operations. Computers & Chemical Engineering, 24, 2259-2276.

Polasek, M., & Mann, J. (2013). Distillate blending optimization. In. White Paper: Honeywell Process Solutions.

Reddy, P. C. P., Karimi, I. A., & Srinivasan, R. (2004a). A novel solution approach for optimizing crude oil operations. A.I.Ch.E. Journal 50, 1177-1197.

Reddy, P. C. P., Karimi, I. A., & Srinivasan, R. (2004b). A new continuous-time formulation for scheduling crude oil operations. Chemical Engineering Science, 59, 1325-1341.

Roslof, J., Harjunkoski, I., Westerlund, T., & Isaksson, J. (2002). Solving a large-scale industrial scheduling problem using MILP combined with a heuristic procedure. European Journal of Operational Research, 138, 29-42.

Ruszczynski, A. (1995). On convergence of an augmented Lagrangian decomposition method for sparse convex optimization. Mathematics of Operations Research, 20, 634-656.

Saharidis, G. K., Dallery, Y., & Karaesmen, F. (2006). Centralized versus Decentralized Production Planning. RAIRO Operations Research, 40.

Saharidis, G. K. D., & Ierapetritou, M. G. (2009). Scheduling of Loading and Unloading of Crude Oil in a Refinery with Optimal Mixture Preparation. Industrial & Engineering Chemistry Research, 48, 2624-2633.

Saharidis, G. K. D., Kouikoglou, V. S., & Dallery, Y. (2009). Centralized and decentralized control polices for a two-stage stochastic supply chain with subcontracting. International Journal of Production Economics, 117, 117-126.

Saharidis, G. K. D., Minoux, M., & Dallery, Y. (2009). Scheduling of loading and unloading of crude oil in a refinery using event-based discrete time formulation. Computers & Chemical Engineering, 33, 1413-1426.

Saharidis, G. K. D., Minoux, M., & Ierapetritou, M. G. (2010). Accelerating Benders method using covering cut bundle generation. International Transactions in Operational Research, 17, 221-237.

Shah, N. (1998). Single and multisite planning and scheduling: current status and future challenges. AIChE Symposium Series, 94, 75.

Shah, N. (2004). Pharmaceutical supply chains: key issues and strategies for optimisation. Computers & Chemical Engineering, 28, 929-941.

Shah, N. (2005). Process industry supply chains: Advances and challenges. Computers & Chemical Engineering, 29, 1225-1236.

Shah, N. K., Li, Z., & Ierapetritou, M. G. (2011). Petroleum Refining Operations: Key Issues, Advances, and Opportunities. Industrial & Engineering Chemistry Research, 50, 1161-1170.

Shaik, M. A., & Floudas, C. A. (2007). Improved Unit-Specific Event-Based Continuous-Time Model for Short-Term Scheduling of Continuous Processes: Rigorous Treatment of Storage Requirements. Industrial & Engineering Chemistry Research, 46, 1764-1779.

Shaik, M. A., Floudas, C. A., Kallrath, J., & Pitz, H.-J. (2009). Production Scheduling of a Large-Scale Industrial Continuous Plant: Short-Term and Medium-Term Scheduling. Computers & Chemical Engineering, 33, 670-686.

Shapiro, J. F. (2001). Modeling the Supply Chain. Duxbury: Pacific Grove.

Stefansson, H., Sigmarsdottir, S., Jensson, P., & Shah, N. (2011). Discrete and continuous time representations and mathematical models for large production scheduling problems: A case study from the pharmaceutical industry. European Journal of Operational Research, 215, 383-392.

Tang, L., & Jiang, S. (2009). The Charge Batching Planning Problem in Steelmaking Process Using Lagrangian Relaxation Algorithm. Industrial & Engineering Chemistry Research, 48, 7780-7787.

Timpe, C. H., & Kallrath, J. (2000). Optimal planning in large multi-site production networks. European Journal of Operational Research, 126, 422-435.

Tosserams, S., Etman, L. F. P., Papalambros, P. Y., & Rooda, J. E. (2006). An Augmented Lagrangian Relaxation for Analytical Target Cascading Using the Alternating Direction Method of Multipliers. Structural and Multidisciplinary Optimization, 31, 176-189.

Tosserams, S., Etman, L. F. P., & Rooda, J. E. (2008). Augmented Lagrangian coordination for distributed optimal design in MDO. Int. J. Numer. Meth. Engng., 73, 1885-1910.

Varma, V. A., Reklaitis, G. V., Blau, G. E., & Pekny, J. F. (2007). Enterprise-wide modeling & optimization--An overview of emerging research challenges and opportunities. Computers & Chemical Engineering, 31, 692-711.

Varvarezos, D. Recent advances in planning and blending operations. In Refinery Optimization: Aspen Technology, Inc.

Velez, S., Sundaramoorthy, A., & Maravelias, C. T. (2013). Valid Inequalities Based on Demand Propagation for Chemical Production Scheduling MIP Models. AIChE Journal, 59, 872-887.

Verderame, P. M., & Floudas, C. A. (2008). Integrated Operational Planning and Medium-Term Scheduling for Large-Scale Industrial Batch Plants. Industrial & Engineering Chemistry Research, 47, 4845-4860.

Verderame, P. M., & Floudas, C. A. (2009). Operational planning framework for multisite production and distribution networks. Computers & Chemical Engineering, 33, 1036-1050.

Wassick, J. M. (2009). Enterprise-wide optimization in an integrated chemical complex. Computers & Chemical Engineering, 33, 1950-1963.

Wilkinson, S. J., Cortier, A., Shah, N., & Pantelides, C. C. (1996). Integrated production and distribution scheduling on a Europe-wide basis. Computers & Chemical Engineering, 20, S1275-S1280.

Wu, D., & Ierapetritou, M. (2007). Hierarchical approach for production planning and scheduling under uncertainty. Chemical Engineering and Processing: Process Intensification, 46, 1129-1140.

Wu, D., & Ierapetritou, M. G. (2003). Decomposition approaches for the efficient solution of short-term scheduling problems. Computers & Chemical Engineering, 27, 1261-1276.

Zhao, X., Liu, W., & Wang, J. (2008). Genetic Algorithm optimizer for blend planning. In Control Conference, 2008. CCC 2008. 27th Chinese (pp. 86-88).