5-2011

A MULTI-COMMODITY NETWORK FLOW APPROACH FOR SEQUENCING REFINED PRODUCTS IN PIPELINE SYSTEMS

Rolando José Acosta Amado
racostaa@utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Rolando José Acosta Amado entitled "A MULTI-COMMODITY NETWORK FLOW APPROACH FOR SEQUENCING REFINED PRODUCTS IN PIPELINE SYSTEMS." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Alberto Garcia, Major Professor

We have read this dissertation and recommend its acceptance:

Chanaka Edirisinghe, Xiaoyan Zhu, Xueping Li

Accepted for the Council:
Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a dissertation written by Rolando Jose Acosta Amado entitled "A MULTI-COMMODITY NETWORK FLOW APPROACH FOR SEQUENCING REFINED PRODUCTS IN PIPELINE SYSTEMS." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Industrial Engineering.

Alberto Garcia, Major Professor

We have read this dissertation
and recommend its acceptance:

Chanaka Edirisinghe

Xiaoyan Zhu

Xueping Li

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

A MULTI-COMMODITY NETWORK FLOW APPROACH FOR SEQUENCING REFINED PRODUCTS IN PIPELINE SYSTEMS

A Thesis Presented for
the Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Rolando José Acosta Amado
May 2011

ABSTRACT

In the oil industry, there is a special class of pipelines used for the transportation of refined products. The problem of sequencing the inputs to be pumped through this type of pipeline seeks to generate the optimal sequence of batches of products and their destination as well as the amount of product to be pumped such that the total operational cost of the system, or another operational objective, is optimized while satisfying the product demands according to the requirements set by the customers. This dissertation introduces a new modeling approach and proposes a solution methodology for this problem capable of dealing with the topology of all the scenarios reported in the literature so far.

The system representation is based on a 1-0 multi commodity network flow formulation that models the dynamics of the system, including aspects such as conservation of product flow constraints at the depots, travel time of products from the refinery to their depot destination and what happens upstream and downstream the line whenever a product is being received at a given depot while another one is being injected into the line at the refinery. It is assumed that the products are already available at the refinery and their demand at each depot is deterministic and known beforehand. The model provides the sequence, the amounts, the destination and the trazability of the shipped batches of different products from their sources to their destinations during the entire horizon planning period while seeking the optimization of pumping and inventory holding costs satisfying the time window constraints.

A survey for the available literature is presented. Given the problem structure, a decomposition based solution procedure is explored with the intention of exploiting the network structure using the network simplex method. A branch and bound algorithm that exploits the dynamics of the system assigning priorities for branching to a selected set of variables is proposed and its computational results for the solution, obtained via GAMS/CPLEX, of the formulation for random instances of the problem of different sizes are presented. Future research directions on this field are proposed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Pipelines have played an important role in the growth of American economy being a widely used transportation method for commodities like water, oil and gas. One of the main ways of transporting oil products is by pipeline networks. A network of 200000 miles of pipelines safely and efficiently supplies America with fundamental commodities for the American way of life. Petroleum pipelines transport 17% of all U.S. freight, but cost only 2% of the nation's freight bill, its operating costs are lower compared to other freight modes that have less capacity and need more resources to be operated[1]. Petroleum products daily consumption in America is estimated in 20 millions of barrels per day. The Association of Oil Pipelines (AOPL) estimated that 68% of domestic shipments of petroleum were by pipelines in 2000. 25% of US inter-city freight is transported using pipelines [40]

Different refined products can be transported using a single pipeline known as a *polyduct*. The products involved in the operation of this kind of transportation system are classified in two types: miscible and non-miscible products [29]. Miscible products are those that can be sequenced consecutively with no contamination making reprocessing necessary, like different grades of gasoline. Non-miscible products on the other hand, are those that should not be sequenced consecutively because product contamination would happen and there would be a higher cost of reprocessing to separate the two products contained in the interface, which is the mix of the two non-miscible products known in the literature as *transmix*. These interfaces represent an additional source of operational cost for the system. During the transportation process, different batches of product are pushed through the system abutting each other. Mechanical separators are seldom used. Products should be sequenced to permit most interfaces to be downgraded from premium to regular products. The challenge is to come up with an optimal sequence of batches of refined products to satisfy the customer demands while optimizing the total operational cost of the system or another operational objective of interest.

Background

The crude oil value chain is divided in *exploration*, *production*, *transporting*, *refining* and *marketing*. In the downstream of the crude oil value chain are the refining and marketing processes. The refining process consists of converting crude oil into finished products. Distributing and selling refined products are Marketing activities. To the light of this description,

[1] Association of Oil Pipelines (www.aopl.org)

the problem under consideration is a Marketing problem in the crude oil value chain. Rejowski and Pinto [33] identify the problem as one in the transportation activities that occur between refineries and depots in the generic petroleum supply chain. The distribution of refined products can be carried out using different transportation modes; nonetheless, pipeline systems provide a very efficient and safe mode of transportation for these products. Transportation and distribution ensure that crude oil will be available in refineries and that products will be distributed through local markets that are spread throughout the world [34]. On land, crude oil and refined products are transported via pipelines, trucks and trains. However, nearly two thirds of the petroleum products in the US are transported by pipelines [85]. Pipelines are the lowest cost transportation method. Once the products reach their destination, which is usually a supply terminal, they are distributed to gasoline stations, airports and homes by tanker trucks. It is convenient for pipeline companies to maximize the utilization of pipelines because of their low operational costs.

Pipelines are the most efficient method to transport crude oil and refined products. Product pipelines ship gasoline, jet fuel and diesel fuel, home heating fuel and kerosene among others, from the refinery to the local distribution facilities. Because of this variety of products being transported through the same pipeline, batching is necessary. The adjoining batches of different products mix where they come into contact. This mixed stream may be sent to the refinery for re-refining, sold as a lower valued product such as a mixture of premium unleaded gasoline with regular unleaded gasoline, or sold as mixture. In any case, there is a cost associated with the sequencing of 2 different products consecutively. Oil is generally propelled through pipelines by centrifugal pumps. Oil moves through pipelines at speeds of approximately 3 to 8 miles per hour. At this rate, it takes from 14 to 24 days to move liquid from Houston, TX to New York City, with 18.5 days the average time[2]. A batch is a quantity of one product or grade that will be transported before the injection of a second product or grade. Batching crude oil and refined products for pipeline transportation has become a more complex task with the proliferation of product qualities, not only refined products but also crude oil of distinct qualities. The new products require more batching and allow less scheduling flexibility making the sequencing problem to become more challenging. They also increase the number of interfaces, and thus require more products to be downgraded from one grade to the next lower grade. New stringent regulations have increased the volume of *transmix* created in the transportation process and consequently, the amount of product that must be reprocessed to meet specifications is also increasing causing a growth in the operational cost of the system. Perhaps that is why this problem has been attracting more attention from the community.

[2] www.aopl.org

Terminals are a critical part of the delivery infrastructure and impact pipeline operations. In some instances shippers on the pipeline or independent operators own the terminals. In other instances the pipeline transporter provides storage services. The proliferation of mandated product grades leads to underutilization of tank age and other assets, creating challenges for any terminal operator and all pipelines alike[3]. The pipeline flow direction can be reversed but in most cases it requires elaborate and costly reworking. The greater the volume being transported on a given day, the faster the product moves.

All levels of decisions arise in the petroleum supply chain: strategic, tactical and operational. In spite of the complexity involved in the decision making process at each level, much of their management is currently still based on heuristics or on simple linear models [10, 18, 24, 34]. Scheduling has a lack of rigorous mathematical approaches to describe the entire refinery operation. Therefore, schedulers usually base their work on experience, heuristics and the use of spreadsheets [34].

Distribution operations are very important in the oil industry supply chain. The optimization of distribution operations in this system is very elusive if we consider its inherent interdependencies that result in a complexity difficult to deal with. The model formulation and a solution methodology, based on decomposition strategies, to deal with large scale instances of this problem are 2 of the most important challenges in Enterprise Wide Optimization for the process industries [13].

Problem Definition

The pipeline schedule defines the product sequence to transport and the lot volumes and associated timing issues, beginning and ending times of each lot pumping and discharging. The schedule enables the maintenance of a feasible inventory level during the entire time horizon, considering settling periods, maximum and minimum tank capacity and satisfaction of client demands [34].

The operation of a multi fluid transportation system consists of determining the order and the way in which the different products are going to be transported to satisfy the demand [1]. Pipeline scheduling is a difficult optimization problem. It is the goal of the pipeline operators to match de demand for products with the physical barrels at each loading terminal [8].

The problem of interest in this research project is concerned with the sequencing of batches of different refined products to be shipped via a pipeline system in order to minimize the total

[3] www.petrostrategies.org

operational costs in a given period of time, while satisfying the considered characteristic operational constraints of the system and the demand satisfaction of the different products at the different market zones within the corresponding delivery time window. The system consists of a refinery producing a set of P products, a set of D depots serving a corresponding market zone where a demand for each product is known and assumed to be deterministic and a pipeline for refined products connecting the refinery and each depot. In the depots and in the refinery, there might be more than one storage tank for each type of product. Scheduling product batches in pipelines is a complex task with many constraints. Producer's production schedules and market demands together with operational constraints forbidding some products to be pumped one after another are all to be considered. Actual inventories available in storage tanks at origin and distribution terminals as well as product batches already in transit to the nominated destination should also be considered [3].

Pipeline scheduling aims to [3]:

- minimize the cost of pipeline operations and keep the pipeline running as close as possible to maximum capacity,
- enhance shipper information about the status of product movements and
- Take advantage of time varying energy costs for pump power.

Contribution

The optimization of operations of supply, manufacturing and distribution activities of a company in terms of costs and inventories presents 3 challenges: modeling of planning and scheduling, multi scale optimization and handling of uncertainties. One of the major issues related with the first challenge is the development of novel mathematical programming models that can be effectively integrated to capture the complexity of the various operations. Providing novel decomposition procedures that can effectively work across large spatial and temporal scales is an issue related with the second challenge [13]. Only one paper reported a decomposition strategy implementation to schedule multiple commodities to be distributed via pipeline. For this problem, decomposition schemes have not been explored. Usually, the problem is solved using a single model solution strategy.

This dissertation contributes by

- Providing a literature review for the problem under consideration which is not available at this point. A time framework of the problem, main authors and their contributions as well as research perspectives for the problem of interest will be presented. Fertile research directions in this field will be proposed

- Introducing a valid novel multi commodity network flow approach for the problem of sequencing batches of refined products to be shipped via pipeline in order to satisfy their demand at market zones while meeting delivery time window constraints. This contributes to the first challenge mentioned in [13].
- Exploring a decomposition based solution methodology. Its implementation will be illustrated and its performance in terms of both solution quality and computational time will be studied.
- Proposing a Branch and Bound algorithm that exploits the dynamic aspects of the system via branching priorities

Grossman et al. [13] state that further research is required to expand the scope and size of planning and scheduling models that can be solved in order to achieve the goal of enterprise wide optimization. And they give special importance to the development of effective decomposition schemes that have the capability of handling large scale problems over geographically distributed sites and over wide temporal scales. These models and methods have the potential of providing a new generation of analytical IT tools that can significantly increase profits and reduce costs. Currently, pipeline operation is based on experience and no computer algorithm is used [18]. Nowadays, the scheduling process is still defined by operator's skills [20]. Cafaro and Cerdá [3] provide a very good description of how a request from an oil company for transportation service during the next coming monthly period is processed.

Organization of Dissertation

A survey of the transportation of refined products problem via pipeline systems is presented in the next chapter. Chapter 3 presents the overall conceptual approach of the proposed modeling approach; Chapter 4 is devoted to present the mathematical model formulation, solution methodology, model extensions and an illustration for a small scenario of the problem. Chapter 5 presents the computation and application of the developed model and the proposed solution procedure to some examples found in the literature. The computational performance of the proposed solution procedure for the mathematical model is discussed. The analysis of results is provided in Chapter 6 and Chapter 7 presents a section for the summary, conclusions and recommendations. Also, at the end of this document, references and a section devoted for appendixes are included.

CHAPTER II

LITERATURE REVIEW

This chapter presents a review of the literature related to the refined products distribution problem via pipeline systems (*RPDPPS*). The papers considered include all where the scheduling of pipelines for multiple refined products transportation is involved. 27 references were considered in the survey as relevant publications. 3 types of publications were surveyed: book chapters (3), conference papers (13) and journal articles (11). Assumptions, optimization criteria, modeling approaches, solution procedures and aspects related to the type of instances solved in each paper are considered. This chapter is divided in 3 sections. The first section presents an overview of the literature of the problem of interest emphasizing in the modeling approach, solution methodology and computational experiments. The second section provides a research perspective for this economically important problem. Conclusions about the conducted survey for this problem as well as suggested directions for future research in this field are presented in the last section.

Problem Overview

The petroleum industry has been a major innovator of Management Science applications. Management Science has been used to develop decision aids in such areas as oil and gas operations, crude oil acquisition, refinery planning, unit process control, refinery scheduling, blending and distribution planning [17]. Regarding the last topic, several research efforts have been made around the pipeline transportation field by many authors.

Camacho et al. [7] present a discrete simulation model of an oil pipeline which main objective obeys scheduling purposes. The paper addresses the operation of a multi fluid transportation system. The order and the way in which the different fluids are going to be transported to satisfy the demand have to be determined. 2 different problems are considered in order to accomplish this task: the determination of the approximate transportation needs, which determines the batch sequence that will minimize the number of interfaces and will cover the consumption needs at each destination node; and an approximate pumping schedule. The second problem involves determining how to set the different pumps and valves at each time interval so that the batch sequence is carried out in a given period of time. Given a pumping sequence and an initial state for the pipeline, the user should be able to simulate with reasonable accuracy the arrival time of the different batches at the terminals, the state of the pipeline at any given moment and the evolution of the level at the terminals, in order to judge whether the pumping schedule fulfills the needs of the terminals. A steady state model is chosen and 2 possible ways of working the simulator are proposed: automatically, all actions are taken by the program in order to optimize the electricity bill; and manually, where the user takes all the necessary actions. The main events are: the arrival of the interfaces at the components, tank levels reaching limits, changes in electrical tariffs, shut down of the system, installation starting up and periodic events. The simulator first reads the data on the files and obtain the initial configuration of the system; second, it adjusts the different parameters of the system manually or using the optimizer; third, it calculates the pressure and flow values throughout the pipeline; fourth, calculates the time in which the next event will take place; fifth, with the obtained time, it calculates the values of the new state of the pipeline; sixth, it returns to the second step. The problem consists of finding out how to set the pumps and valves at each time interval in order to deliver the products to the terminal at appropriate rates with minimum electricity costs. The costs function consists of two components given in terms of a state vector consisting of a pair volume-time. The first component represents the minimum cost from the origin to a given node and the second component estimates the minimum cost from the given node to a goal node. The optimization algorithm works in two steps. The first one optimizes the second component of the objective function and the second one runs the simulation with this information. The simulator and the optimization algorithm have been included in a program developed for CAMPSA. The program can be applied to any pipeline transportation system with only one entry node.

Hane and Ratliff [15] examine the problem of sequencing the input of commodities to a pipeline so that a surrogate function of pumping and maintenance costs is minimized. The main contribution of the paper is the formalization of this industrial problem. The pipeline problem *P* is defined by the physical structure of the pipes and the static set of orders. The pipeline structure is modeled as a directed network *G(V,E)*, where *V* is the set of nodes and *E* the edges. Each edge has an integral volume which represents the volume between the end nodes of the edge. The nodes of *V* correspond to the sources, destinations and junctions in the pipeline system. The set of orders *O* defines the commodity, input node, delivery node and integral amount to be delivered for each order. They assume that there is at least one flow in any time period and the rate of flow is constant. It is assumed that momentum propagates instantaneously through the pipe which implies that if an amount of product *x* enters the pipeline, another amount of product *x* must exit it. No time window constraints are considered as part of the problem. The pipeline is operated in a cycle of inputs basis. The backfill algorithm that is used to determine the pipeline contents at the beginning of the steady state is introduced. The proposed model seems to be for a strategic decision level. Costs due to mixing can be captured in the objective function, but there is no means to handle product loss or migration of fluid from one commodity to another. One goal of the sequence is to reduce the variance in energy demand. Because of an apparent intractability of the cost function, a surrogate cost function is used. Another goal of the sequence is to minimize the costs of all required stoppages. The sequencing algorithm as well as a branch and bound algorithm are presented providing their mathematical background. Finally, computational results of the implementation of their algorithm are provided. The mathematical formulation of the model is not provided though.

Sasikumar et al. [39] address the pipeline schedule generation problem to generate a pumping schedule for a single source multiple destinations pipeline system to distribute multiple products. A schedule for this problem is understood as the sequence of products to be pumped for the period specified, along with the quantity and how it should be distributed among the market zones. The task of the system is to generate a good pumping schedule for a period of about a month, based on the information of available supply and required amounts for each product at the different zones while meeting the constraints of the system. The problem is considered in the type of a resource scheduling problem. A knowledge based heuristic search approach is proposed. The search space is defined based on the concept of a *move*, which is the choice of the next batch to pump and has 4 components: the product to be pumped, the quantity to be pumped, how the product is to be distributed among the destinations and the pumping sequence being followed. The state representation captures the scenario of all the locations and the pipeline just before a new batch is pumped. This state is given by the projected inventory map of all products at all locations, the current content of the pipeline and the current time. A fixed width search is used to explore the search space. The domain constraints are applied to each list of nodes and the resulting nodes are evaluated using a heuristic function. The evaluation function considers a weighted sum of a number of factors in assessing the schedule so far such as the cost of the schedule so far, the predicted stock level at all locations for all products as per the current schedule, how much of the requirements of the locations have been satisfied, penalty for any shutdown in the pipeline and penalty for violating plug limits. The goal is not to optimize the system performance but to generate a feasible pumping schedule for one month instead. They use the information of a system in India, which has a 500 kilometers pipeline, one refinery, 3 market zones and distributes 4 products to implement a computational experiment.

Rejowski and Pinto [29] consider a system composed by a petroleum refinery, a multi-product pipeline and several depots that are connected to local consumer markets. The refinery must distribute P petroleum products between D depots connected to a single pipeline, which is divided into D segments. The depots have to satisfy requirements determined by local consumer markets. A mixed integer and linear programming formulation for the system is introduced for the simultaneous optimization of systems with multiple depots. A uniform discrete time representation is used. The results generated by this model are the inventory level profiles for all products at the refinery in all pipeline segments and at the depots along the distribution horizon. The pipeline is divided in segments and the segments are sub divided in packs. The model is implemented in a real world instance located in Brazil where one pipeline distributes 4 products to satisfy their demand at 5 market zones. The model seeks to optimize inventory costs at the refinery and at the depots as well as pumping costs and product transition costs. A single 3-day horizon time scheduling instance for the problem is presented for which a 4.7% relative optimality gap is achieved using GAMS.

Milidiú et al. [21] propose a model for pipeline transportation of petroleum products with non-cyclic orders: the Pipeline Transportation Optimization Problem (PTOP). PTOP represents a pipeline system through a directed graph where each node represents a location and each directed arc represents a pipeline with a corresponding flow direction. Both, ordered volumes and pipeline capacities are integers. The term batch is used to denote the amount of product that corresponds to a given unitary volume order. These batches cannot be split during transportation. Each batch is defined by its initial position and its associated destination node. Batches with fixed destination nodes are usually called proprietary batches. They assume all batches to be proprietary, that is, no fungible products. The PTOP model assumes that fluids are incompressible, location storages are unlimited, all batches are proprietary, batch volumes are unitary and batches cannot be split. The pipeline system is represented by a directed arc where the arcs model the pipes and the nodes model the locations. The concepts of further order and non-further orders are presented and used to introduce the concept of further batch and non-further batch. Further orders are not necessarily satisfied at the end of a feasible pumping sequence and non-further orders are satisfied during the pumping sequence. Any solution to this model generates a discrete sequence of states, where the positions of all batches are well defined. A solution for the model is a sequence of elementary pipeline operations, EPO. An objective function is defined for each EPO considering the pumping costs. A proof of PTP being a NP-hard problem is provided. The proof is performed by showing a polynomial reduction from the vertex cover problem to PTP. A feasible solution is defined as a pumping sequence that delivers all batches corresponding to the non-further orders. The problem of finding a feasible solution to PTOP is referred to as PTP. A Batch-to-Pipe Assignment –BPA- algorithm to test the feasibility condition is proposed. BPA runs in polynomial time. The Unitary Batch-to-Pipe Assignment Algorithm is introduced. The algorithm performs 6 main steps. In the first step, a weighted shortest path for each pair of nodes is constructed. Step 2 constructs a weighted bipartite graph considering each batch, each pipeline position, the minimum cost of transporting a given batch through a valid route. Step 3 checks the feasibility of the solution and stops the algorithm if the solution is infeasible. If the solution is feasible, step 4 is performed. In this step, the minimum valid route for each batch is determined. Step 5 constructs the corresponding dependence graph and step 6 calls the sequencing procedure to construct a feasible solution. A batch route is defined as the chronologically ordered sequence of arcs traversed by a batch when the corresponding pumping sequence is applied. The total cost of the given sequence can be expressed as a function of the routes, the initial positions and the final positions of all batches. The sequencing procedure receives the following information for each batch: a valid final state, a valid route consistent with the given final state and the corresponding dependence graph. For each batch, the route and final position may be changed by sequencing. If the graph is acyclic in every iteration, sequencing selects a source node from the graph and pumps through it every batch whose route contains that node. After that, the node is removed from the graph and no longer used.

Rejowski and Pinto [30] address the problem in which a refinery must distribute P petroleum products among D depots connected to a single pipeline, which is divided into D segments. The depots have to satisfy requirements determined by local consumer markets. A mixed integer and linear programming formulation for the system is proposed. The results generated by the model are the inventory level profiles for all products at the refinery, at all pipeline segments and at the depots along the distribution horizon. Two mathematical models are presented. First, a model that considers packs of equal volumetric capacity is considered. For the second model, this assumption is relaxed. An integer cut is proposed in order to reduce the combinatorial search in both models. The constraint is based in the minimum number of times that a depot d must receive product p from the pipeline along the time horizon. 2 examples with a time horizon of over 3 days, one for each model, are presented and they report the use of GAMS to solve them. In both cases, they consider 1 refinery, 5 depots, one pipeline and 4 products. A major challenge in the problem is to monitor product content in the pipeline that is subject to intermittent operation.

Milidiú et al. [22] describe the liquid pipeline transportation problem, more specifically, multi commodity liquid pipelines where more than one petroleum derivative may be transported. The main components of a pipeline network are operational areas such as distribution centers, ports or refineries; and the pipeline segments. These areas are connected by one or more pipeline segments. Interface restrictions, reverse flows, storage constraints at market zones, operational flow rates and production/demand constraints are considered. A PDDL model is presented. PDDL stands for Planning Domain Definition Language and focuses on expressing the physical properties of the domain that is considered in a given planning problem [41]. The purpose of the pipeline schedule is to elaborate a sequence of segment content movements such as there are available products at the areas where it is demanded while the constraints corresponding to tank levels are met for refineries and ports. Each pipeline segment is modeled as a block stack which must keep its size constant.

Cafaro and Cerdá [3] address the problem of establishing the optimal sequence of new slugs injections in the pipeline, their initial volumes and the product assigned to each one in order to meet product demands at each depot in a timely fashion, keep inventory levels in refinery and depot tanks within the permissible range all the time and minimize the sum of all pumping, transition and inventory carrying costs. At the same time, variations in sizes and coordinates of new/old slugs as they move along the pipeline as well as the evolution of inventory levels in refinery and depot tanks are tracked over the time horizon. A novel non-discrete MILP formulation for the optimal scheduling of multiproduct pipeline systems is proposed. The problem goal is to establish the optimal sequence of new slug injections in the pipeline, their initial volumes and the products assigned to each one in order to meet the product demand at each depot in a timely fashion, keep inventory levels in the refinery and depot tanks within the permissible range all the time, minimize the sum of all pumping, transition and inventory carrying costs, variation in sizes and coordinates of new-old slugs as they move along the pipeline and the evolution of inventory levels in refinery and depot tanks. A MILP continuous time approach for the scheduling of a single pipeline transporting refined petroleum products from a unique oil refinery to several distribution terminals is proposed. This formulation neither uses time discretization nor division of the pipeline into a number of single product packs. To illustrate their approach, they report the solution of 2 real world case studies first introduced by Rejowski and Pinto [30]. This approach over performs Rejowski and Pinto’s [30] results for both examples in terms of solution quality, number of binary variables, number of constraints and CPU time.

Magatão et al. [18] focus on the short term scheduling of activities in a specific pipeline system. It connects a harbor to an inland refinery. The pipe conveys different types of commodities which are oil derivatives. It is possible to pump products either from the refinery to the harbor (flow procedure) or from the harbor to the refinery (reflow procedure). The pipeline operates uninterruptedly and there is no physical separation between successive products as they move in the pipe. Consequently, there is a contamination area between miscible products: the interface. Additional operational costs are generated out of these interfaces. A decomposition strategy to address a large-scale scheduling problem that is found in a real-world pipeline scenario is proposed. Their work is focused on the short-term scheduling of activities in a specific pipeline system. It connects a harbor to an inland refinery. The pipeline is 93.5 km length, can store a total volume of 7314 cubic meters and connects a refinery tank farm to a harbor tank farm going along regions with 900 meters altitude difference. The pipe conveys different types of commodities (gasoline, diesel, kerosene, alcohol, liquefied petroleum gas, jet fuel, etc.) which are oil derivatives. It is possible to pump products either from the refinery to the harbor or from the harbor to the refinery (this is called reflow procedure). The pipe operates uninterruptedly and there is no physical separation between successive products as they move in the pipe. There is a contamination area between miscible products: the interface. Some interfaces are operationally not recommended and a plug can be used to avoid specific interfaces, even though, plug inclusions increase the operational cost. The core methodology applied is an MILP model with uniform time discretization. The computational complexity is considered and an optimization structure is proposed to decompose the problem. One main model, a tank bound model and an auxiliary routine are introduced. The tank bound model accounts for the minimization of the cost variable that is composed by the tank changeovers, and the specification of logical conditions involving product availability. The satisfaction of demand requirements within an operational range, the operational limits for tank volume, the siphoning of the tanks used to supply demand requirements and the tanks that should be used to satisfy pumping activities are all aspects modeled in the tank bound model. The auxiliary routine considers the minimum time horizon to complete the entire pumping procedure, determines the end procedure parameters. Also, the limits that help to narrow the main model search tree are established by the auxiliary routine. The main model defines the operational cost minimization. The optimization structure must determine the ideal flow rate policy during a limited time horizon. The conditions that at least one batch has to be pumped at the initial time and each product is pumped only once throughout the scheduling horizon are modeled as constraints in the main model. Also, it takes the temporal limits determined by the auxiliary routine and sets up binary variables to determine whether or not a given product starts being pumped at a given time as part of a reflow procedure or flow procedure. More detailed aspects are also modeled such as the time interval that the pipe empties a tank, avoidance of overlaps between batches, product flow rates, demand requirements, pipeline flow rate at each discretized time and more operational details. One instance of the problem is presented where 4 products are sent in a reflow procedure and 4 products are sent in a flow procedure. Flow rates, demanded amounts and plug needs are considered. Also, electric cost variations are modeled. The model is solved to optimality using LINGO/PC Release 8.0.

Cafaro and Cerdá [4] introduce an efficient multi period MILP continuous approach to the DPSP based on the formulation of Cafaro and Cerdá [3] for the static pipeline scheduling problem. This approach is capable of optimally updating the sequence of pipeline product injections over a rolling horizon. The problem goal is to dynamically establish/update the optimal sequence of pumping runs over a multi period time horizon in order to meet every product demand at each period in a timely fashion, maintain the inventory level in refinery and depot tanks within the permissible range and minimize the sum of pumping, transition and inventory carrying costs. New problem variables are added to the original mathematical formulation presented by Cafaro and Cerdá [3]. By considering a multi period planning horizon, the new formulation is capable of handling multiple due dates for the product deliveries to different distribution terminals which are supposed to occur at period ends. New parameters are considered to account for the initial and final time of each periods of the set in which the horizon planning is divided. Also, the demand of each product at each depot before the end of each period of the horizon planning is considered. A new binary variable is defined to denote whether a given pumping run is completed inside or at the end of each period. New constraints control the completion time period of a new pumping run and enforce that the total amount of a given product dispatched from a given terminal to the local market permits to meet the demand of the product from the first period to a given period. In order to illustrate the advantages of the proposed dynamic pipeline scheduling approach, the real-world example introduced by Rejowski and Pinto [30] was solved but this time a much longer multi period horizon and multiple delivery due dates were considered. Four weekly periods is the span of the planning horizon. Product demands at depots $D_1 - D_5$ to be satisfied at the end of periods $t_1 - t_4$ Demand data for the subsequent time intervals $t_5 - t_7$ is still unknown at the time of developing the static pipeline schedule for the initial horizon $t_1 - t_4$. This data becomes available as the four period horizon rolls. They assume similar demand profiles and refinery outputs for the next 3 periods of the horizon planning. Once changes in the demand patterns are made, the original schedule also changes significantly. Results show that the sequence of pumping runs finally executed over the horizon looks quite different from the one found through a static pipeline scheduling technique. Pumping runs become shorter and increase in number. The scheduled pipeline idle time decreases.

Rejowski and Pinto [31] address the short term scheduling problem where a refinery must distribute P petroleum products among D depots connected to a single pipeline, which is divided into D segments that may represent decreasing diameters. In the refinery and in the distribution depots, several tanks store the same product, although at most one of these is connected to the pipeline at each time. The objective is to generalize and improve the efficiency of their MILP formulation proposed in [30] by adding special and non-intuitive practical constraints, which minimizes product contamination inside the pipeline segments and the resulting model is analyzed in terms of computational performance and solution quality. They report that the new formulation find the optimal solution with a higher value when compared to a feasible one of the respective problems without the new features. The system reported in this work is composed by an oil refinery, one multiproduct pipeline connected to several depots and to the local consumer markets that must be fed with large amounts of oil products. Case studies are reported for 3 scenarios of low, medium and high demand patterns and their results discussed.

Rejowski and Pinto [32] develop a hydraulic formulation for pipeline scheduling. This paper addresses the simultaneous multiproduct pipeline scheduling and hydraulic operation. The formulation is based on a continuous time representation that handles variable flow rates in the pipeline. The system under consideration is the same described by Rejowski and Pinto [29, 30]. The hydraulic behavior depends on the sequencing of products and their allocation inside the pipeline, the flow rate variations, the topographical profile of the pipeline and diameter variations. The MINLP formulation is based on the formulation presented by Rejowski and Pinto [31]. Temporal and refinery constraints are introduced. The temporal constraints must satisfy the operational time horizon and the initial and end instants along the product transfer operations. Mass balances and volumes are also bounded by additional constraints. A new set of pipeline scheduling and depot constraints is also introduced. The pipeline operation is expressed by disjunction that is composed by two additional terms that relate the flow rate variations and the time interval durations. This linear disjunction is transformed into mixed integer constraints. Temporal variables are disaggregated into two parcels, the first one regarding the pipeline operation and the second one considering the time that the pipeline remains idle. A new set of constraints enforces the speed of products to take positive values. The hydraulic model is described by disjunction as well. A friction factor follows a constraint where a logical binary variable multiplies an exponential that depends on the physical properties of the product. The friction losses for each pack of product take into account the friction factor, the pipeline internal diameter, the pack extension and the pipeline flow rate. All this is considered in a new constraint. Energy balance and power consumption are also controlled with new sets of constraints. The examples presented are based on the instance introduced by Rejowski and Pinto [30]. This new approach is reported to result in better objective function values and also better accuracy. In conclusion, the proposed MINLP approach showed better results than a previous MILP.

Magatão et al. [19] consider the problem involving the short term scheduling of activities in a specific pipeline, which connects a harbor to an inland refinery. The problem topology is the same as in Magatão et al [18]. The task is to specify the pipeline operation during a limited scheduling horizon, providing low cost operational procedures and, at the same time, satisfying a set of operational requirements. The optimization structure presented in Magatão et al. [18] is also used in this work with one fundamental difference: the main model is based on a combined CLP-MILP approach. In the former approach, the main model was just based on an MILP formulation and it can demand a computational effort from minutes to even hours. A set of high level modeling structures was created in order to formulate the CLP-MILP modeling statement and afterwards CLP and MILP equivalent expressions could be automatically derived. The CLP-MILP model is composed by both CLP and MILP formulations, which are iteratively invoked. MILP is used to establish a continuous time scheduling model that enhances the traditional CLP search mechanisms by providing relaxations to the CLP model during the search procedure. Each constraint is formulated as part of a constraint programming model and as part of a mixed integer model. A real life example involving the pumping of 4 products from the harbor to the refinery followed by other four pumped from the refinery to the harbor is considered. A pure MILP, a pure CLP and a combined CLP-MILP approach are considered and numerical comparison amongst three different main model versions is presented. Computational effort demanded by the CLP model is greater than the one demanded by the MILP and the CLP-MILP approaches by orders of magnitude. Both, the MILP and the CLP-MILP approaches demanded a reasonable computational effort. In order to further investigate the computational effort trend presented by the main model, some hypothetical problem instances were also tested. Such instances do not necessarily represent typical operational scenarios. The main goal was to test MILP, CLP and CLP-MILP approaches in theoretically more time consuming problem instances. The task of this work was to predict the pipeline operation during a limited scheduling horizon, providing low cost operational procedures and attending a series of operational requirements. The scheduling of operational activities has to take into account product availability, tankage constraints, pumping sequencing, flow rate determination and a variety of operational procedures.

De La Cruz et al. [9] model and solve the problem of petroleum products distribution through pipeline networks using two techniques: a heuristic method and mathematical programming. The problem of polyduct pipelines is considered as to plan the way different products are temporarily transported from source nodes to demand nodes, passing through intermediate nodes. Time window constraints must be satisfied. A solution to a simplified problem of the optimal distribution of products through pipeline networks using two methods is presented. The two methods are: multi objective evolutionary algorithm and MILP. A simplified model of an actual network is considered with the nodes corresponding to a set of sources, set of sinks or receiving terminals and a set of intermediate connections serving as receiving and delivering points with storage capacity. The goal is to minimize as much as possible the time in which the demand is satisfied as well as the product changes produced in the polyduct. In the heuristic method, the coding of the topology of the network is used via a matrix having as entries the distance among the nodes. Bidirectional links are acknowledged with the symmetry of the matrix. A solution to the problem is given by the kind of packet sent by every source or interconnection node at every instant. The information coding is kept easily in a structure where every row is a cell associated with a node, and within every row, there are as many columns as connections departing from this node. The value of the gene acts as entry to get the product associated with it. A uniform cross point function is used to select randomly the crossover points for each pair of individuals of two genetic populations. The algorithm begins with the creation of an initial population and then some repairs are applied to these individuals so the objective functions are evaluated and the population is ranked using the priority parameter. A dominance matrix is built to keep the relation between each pair of individuals. The population is then divided in several groups: a group with individuals that are not dominated, the group of individuals that remain after eliminating, the individuals of the former group and so on. The individuals in a group are given the same fitness. With the fitness, the MOEA selects the parents for the recombination process and applies the genetic operators to obtain a new population for the next generation. A concrete network was solved using MOEA, MILP and a Hybrid approach and the results were compared. The topology of the network is as follows: 4 products, 2 sources, 3 sink nodes, 2 intermediate nodes, 6 unidirectional edges and 1 bidirectional. Better results were obtained for the hybrid approach where both solvers MOEA and MILP were run in parallel and the solutions obtained from the MILP were used as immigrants into the MOEA.

Relvas et al. [34] addresses the problem of pipeline scheduling and inventory management of a multiproduct distribution oil system. The pipeline schedule defines the product sequence to transport and the lot volumes and associated timing issues, beginning and ending times of each lot pumping and discharging. The schedule enables the maintenance of a feasible inventory level during the entire time horizon, considering settling periods, maximum and minimum tank capacity and satisfaction of client demands. The focus of this work is the operation at the distribution centers. A MILP model to count for the pipeline scheduling and inventory management distribution centers is developed where an extension of the issues raised by previous works is provided and the detailed supply of client demands with daily requirements is considered. Inventory management is not considered to have been studied in previous works. The process involves unloading oil derivatives from the pipeline to the respective distribution center's tanks and then making them available to the local market. There is only one pipeline and only one lot of any product is arriving at each moment. Each tank can assume three different states (at any given moment): loading from pipeline, full and performing settling and approving tasks, unloading for clients. Therefore, the problem not only relies on the scheduling but also on the tanks' inventory management. On the other hand, clients provide a monthly plan on their demands that are to be satisfied on a daily basis. A continuous time MILP model is proposed based on the mathematical formulation of Cafaro and Cerdá [3]. The main differences between their work and the work in [3] rely both on the system studied and the modeling of market behavior and distribution center internal dynamics. The constraints of the model consider lot sequencing, relation between volume and pumping duration, forbidden sequences, upper and lower volume coordinates of a given lot i, pipeline end tasks, product allocation constraints, choice of lot volumes, overall volume balance to the pipeline ends while injecting lot i', initial conditions inside the pipeline, inventory control at the distribution center, client demands, auxiliary conditions. A model extension is introduced to consider client demands on a daily basis in order to build up a more rigorous model that describes real world internal operations in a distribution center. An objective function with multiple optimization criteria is introduced that maximizes the total working time of the pipeline, the amount of transported products, the inventory at the end of the time horizon and penalizes solutions where the number of lots that participate in the settling period is not the maximum possible. A real scenario analysis and computational results is provided for six different oil derivatives, one refinery and one distribution center.

Maruyama et al. [20] addresses the development of a simulation model for the operational decision-making of scheduling activities in a real-world pipeline network. The proposed simulation model is used with a short term scheduling optimization package that provides the scheduling to be simulated. This is accomplished by using a discrete event simulation model implemented in EXTEND where a scheduler generates events at times provided by the optimization package. Each event carries out information about different batches, which are characterized by attributes such as type, route (source, pipe, and destination), volume and flow rate for each product to be transferred. These attributes allow calculating the inventory level at different areas. The considered scenario involves 9 areas, 3 of them are refineries, 1 harbor which either receives or sends products, and 5 distribution centers. The scenario includes 15 pipes, each one with a particular volume. Some of them have their flow direction reverted according to operational requirements. Each product presents a specific tank farm according to the considered area. More than 10 oil derivatives can be transported. Each discrete event corresponds to a pumping start. It carries out information about a batch characterized by attributes such as type, route, volume and flow rate of each product to be pumped. Pumping is accomplished at constant flow rate which determines a linear inventory change. The simulation model has 3 kinds of blocks: scheduler, tank and pipe block. The scheduler generates events at particular times (provided by the optimization package) and it sets event attributes according to information stored on a database. The attribute type represent one of ten possible oil derivatives that flow in the network at fixed rate given by flow rate attribute. The attribute volume is the amount of product in a batch. The attribute route contains a well defined path from a source to a demand area considering all necessary pipes. Each area contains an aggregate storage for each product. In this case, the level of an aggregate tank is subject to three simultaneous behaviors: production, demand and transport. Production and demand fills and drains tanks respectively, while transport may increase or decrease the tank level according to its role (source or destination). All level changes are linear, since production, demand and transport are assumed to have a constant flow rate. The initial conditions for the tanks are their level and storage capacity. Regarding the pipes, each one is modeled as a FIFO queue that stores and releases events (products pushed into the pipe) according to new products arrival. The simulation results generated were obtained for a scenario of 81 batches transferring about 8 products in a time horizon of 20 days.

Relvas et al. [36] present an improved version of the work by Relvas et al [35] and studies the problem applied to the system described. The system under study comprises a refinery, a pipeline and a tank farm that works as a distribution center. The objective of the problem is to find both a pipeline schedule and inventory management plan at the tank farm such that all clients' demands are fulfilled while guaranteeing that all the quality and approving tasks are carried out under their respective constraints. The mathematical model is optimized under the desired objective, which can be either economical or operational. The model building considerations include time and volume scales, associated pipeline stoppage, product, sequences, the daily client information, the tank farm representation and the settling period. The pipeline is viewed as a volume axis, where the origin is the refinery and the destination is the tank farm. The product sequence needs to answer the product needs at the tank farm. It must not contain forbidden sequences between pairs of products, due to quality aspects. The daily client information about demand is transformed from a discrete representation into continuous time scale information. An aggregate tank is used to represent the group of tanks for each product in the mathematical model. A usual procedure at a tank farm for each new batch is to settle for a certain period. This is either for batch quality improvement or to accomplish a set of batch control and approval tests. The settling period is modeled varying with the product gaining closeness to reality and maintaining the solution space. The objective function to be used may be operationally or economically oriented. Behind some operational objectives, there are also economic issues represented, such as flow rate minimization. The minimization of the difference between the total amount of products transported by the pipeline and the total amount of outputs to clients and the maximization of the total pumping time is sought. A survey on situations that may occur in the system in this study is discussed as well as ways of modeling these situations using the MILP model proposed are analyzed. Variation on clients demands, imposition on product sequence, unpredicted pipeline stoppages, batch volume modifications, flow rate adjustments and variation on maximum storage capacity are considered. The first 4 are simple to address and some considerations are outlined for situations 5 and 6. The mathematical model and respective rescheduling framework presented is applied to a real world problem at the CLC – Companhia Logística de Combustíveis, which is a Portuguese oil products distribution company. All scenarios were run using as stopping criteria either a maximum resource time of 7200 CPU seconds or a final solution within a tolerance of 5%. The improvement of this work with respect to the work of Relvas et al. [35] is an extension to the model in order to enable the use of a variable settling period by product, variable flow rate and pipeline stoppages. The authors also proposed a novel procedure to account for reactive scheduling, which enables the decision makers to obtain revised schedules that take into account unexpected events. The rescheduling over the original operational plan can be either performed before or during the time horizon and can accommodate several stand alone or combined perturbations in a single revision. The results obtained reveal that the model is suitable to develop either an initial plan, given the initial conditions of the system or rearrange any current schedule in order to accommodate unexpected changes. The reactive schedule procedure is built in a way that the minimum changes to the previous schedule are obtained.

Neves Jr. et al. [27] address the problem of scheduling decisions within pipeline networks in a particularly complex scenario involving 3 refineries, 1 harbor which either receives or sends products and 5 distribution centers. In addition, it includes 15 pipes, each one with a particular volume. The nodes are connected by various pipes but the list of products that can be pumped by a specific pipe is limited. Some of the pipes involved in the system can have the flow direction reverted, according to operational procedures. More than 10 oil derivatives can be transported. A decomposition approach is proposed to address the problem based on three key elements of scheduling: assignment of resources, sequencing of activities and determination of resource timing utilization by these activities. A preprocessing block (heuristic procedure) takes into account production and consumption functions and typical batch volumes in order to determine a set of candidate sequences of pumping. The preprocessing procedure indicates time windows to the established sequences. The preprocessed data are used by a continuous time MILP model, which determines the operational short-term scheduling for the entire pipeline network. The previously determined time-windows should be respected in order to keep inventory management issues within operational levels. The model considers the pumping route, – source or pumping origin, pipes and destination –, volume and flow rate of each product from a source. The model considers also the seasonal cost of electric energy and a series of operational requirements. The decision variables determine the exact time that a pumping procedure of a batch is started and finished from a node through a specific pipe. Other continuous variables determine the time that a destination node starts to receive and finishes to receive a product. Binary variables were used to enforce seasonality conditions of electric energy. The objective function is weighted by operational cost factors. Specific constraints are proposed to take care of inventory management. The preprocessing unit indicates time-windows to the demanded batches. Some time window violations can be accepted either in the pumping origin or at the final product destination. A specific set of constraints were developed to enforce the possibility of flow reverse operation in a subset of pipelines. The optimization structure was successfully tested in industrial size scenarios where 6000 variables and 20000 constraints were involved. This approach has allowed that a month planning of production and consumption be detailed in short time scheduling operations within the considered pipeline network.

Neves Boschetto et al. [26] address the problem of developing an optimization structure to aid the operational decision making process of the scheduling activities in a real world scenario of a pipeline network. Based on key elements of scheduling, a decomposition approach is proposed using an implementation suitable for model increase. Operational insights are derived from the obtained solutions which are given in a reduced computational time for oil industrial-size scenarios. The proposed approach is compared to the previously developed work in [27] in terms of complexity and computational performance. The considered scenario involves 13 areas including 4 refineries and 2 harbors, which receive or send products through 7 distribution terminals. 29 multiproduct pipelines with particular volumes are used to transport more than 14 oil derivatives in this network. The decomposition is based on the three key elements of scheduling: assignment of resources, sequencing of activities and determination of resource timing used by these activities. A resource allocation block takes into account production and consumption functions and typical volume of batches in order to determine a set of candidate sequences of pumping. The pre-analysis gathers information provided by the resource allocation and calculates a series of temporal and volume parameters (bounds). These bunds provide a preliminary indication about scheduling feasibility. Then, the Pre-Analysis pro-processed data are used by a continuous-time MILP model, which determines the operational short term scheduling for the pipeline network. This model considers the pumping route, volume and flow rate for each product from a source. A novel computational procedure is proposed: Pre-Analysis. This procedure uses information provided by the resource allocation unit to calculate a series of temporal and volume parameters in order to provide structured sequences in a reasonable computational time. As an output, the pre-analysis specifies the precise volumes to be pumped and received in a destination node, the minimum time that a destination node could start to receive and could finish to receive a product. Operational constraints are addressed by the MILP model with a continuous time approach. Variables determine the exact time at which a pumping procedure of a batch is started and finished from a node through a specific pipe, the time at which a destination node starts to receive and finishes receiving a product. In order to determine the values of these variables, parameters obtained from the pre analysis are used. In particular, the pre analysis unit indicates the minimum pumping and receipt time of a batch. The model can deal with seasonality conditions of electric energy. Specific constraints were created to deal with inventory issues. The proposed structure can be used to identify system bottlenecks and to test new operational conditions. Computational time has remained at few CPU seconds.

Moura et al. [24] address the problem of how to schedule all individual pumping operations in order to fulfill market demands and store all the planned production. Each pumping operation is defined by origin and destination tanks, a pipeline route, start and end times, a specific product and its respective volume. The operations must obey all constraints over the given time horizon. The system under consideration is actually a subsystem of a bigger network of pipelines owned by PETROBRAS with around 30 interconnecting pipelines, over 30 different products in circulation, about 14 distribution depots which harbor more than 200 tanks, with a combined capacity for storing up to 65 million barrels. Individual pumping operations have to be scheduled given the daily production and demand of each product at each location in the network, over a given time horizon. An operation is defined by specifying information about the product, volume, route, origin and destination tanks, as well as start and end pumping times. The main goal is to find a solution that respects all operational and physical constraints of the network, as well as that uses stocks and productions to satisfy all local demands, while storing away any remaining production. The complete problem was solved using a hybrid approach that combined a randomized constructive heuristic and a constrained programming model. Two solution stages are presented: first, a constructive heuristic, called the planning phase, is introduced and it is responsible for creating a set of delivery orders. This phase must guarantee that all delivery orders satisfy local market demands and the excess of product will be correctly stored away. The second stage is the scheduling phase that takes the set of delivery orders generated in the planning phase and sequence the pumping operations at the initial pipeline in each route present in a delivery order as well as determine the start times of each of the pumping operations, while ensuring that no network operational constraint is violated at any time. The delivery orders are generated using a randomized constructive heuristic designed based on tacit knowledge from PETROBRAS. 3 steps are followed to generate delivery orders incrementally: first, randomly select a local product demand in any depot giving higher priority to demands that must be fulfilled earlier in time; second, randomly choose depots that could supply volumes of the required products, as well as the routes that these volumes should traverse and third, select origin and destination tanks, setting order volumes accordingly. Also, set order deadlines so as to guarantee demand fulfillment. The planning phase ends as soon as there are no more demands to choose from. The scheduling phase must determine the pumping parameters in order to meet all delivery order deadlines, also taking into account the network operational constraints or prove that the present set of delivery orders is not feasible. The CP model is divided into two steps. The first one deals with the sequencing of delivery orders, generating time intervals for the start of the respective pumping operations. A second simpler model determines the number of pumping operations for each delivery order as well as the start time for each operation. Different types of search strategies were tested for solving both the sequencing and the scheduling models. The currently implemented version combines a backtracking mechanism with a special variable ordering being divided into three consecutive parts: disjunctive components determination, adaptive backtracking and time assignment. 4 real field instances to test the model were used; all share the same network topology of 14 depots, 29 pipelines, 32 different product types and 242 tanks distributed among the depots. Pipeline volumes range from 30 to 8000 cubic meters and most of the tank capacities are between 4000 and 30000 cubic meters.

Moura et al. [25] propose a new algorithm for generating feasible solutions for a very large pipeline planning and scheduling problem, considering most of the hardest real world constraints. The approach has 2 phases: the planning phase, implemented as a constructive heuristic that generates orders, representing necessary transfers between two depots; and the scheduling phase, a constraint programming model that is used to assign time intervals to orders. The resulting algorithm, suitable for dealing with large instances, generates more reliable pumping plans and can also be used to validate production and demand scenarios. A network with 4 depots interconnected by 5 pipelines is considered. Each depot has its own tank farm. Each tank contains an initial volume shown in standardized units. 3 products are considered. A solution is defined as a set of continuous and no preemptive pumping operations defined by the type of product, volume, route, origin and destination tanks as well as start and end pumping times. The main goal is to find a solution that satisfies both all operational as well as all production and demand constraints. The problem is divided in two parts. The first part, planning and routing, aims to satisfy all productions and demands by creating a set of orders that specify routes and volumes. The second part, sequencing and scheduling, defines the sequence and exact times for pumping operations at depots, including those special operations used to store production and extract demands. The first part is handled using heuristic strategies and the second part is solved using a constraint programming model. The planning phase defines a set of orders necessary to satisfy all products demands within the time horizon. The developed heuristic incrementally builds a set of orders in a randomized constructive way. For any order, its products, volume, origin depot, destination depot, origin tank, destination tank, route and due date must be determined. These characteristics are determined sequentially. The planning phase ends when there are no more pairs of product and destination to be chosen. The scheduling phase must gather additional information to control the size of the model, incorporating special structures geared to effectively explore the search space. The proposed CP model explicitly takes advantage of the problem's diverse structures, while also providing the flexibility to consider new operational requirements or implement new searching heuristics. The model comprises 2 different CP perspectives, containing both specific variables and constraints to deal with the corresponding structures they focus on. Constraints are formulated to guarantee the satisfaction of production and demand orders, to represent the pipeline as two time ordered operations sequences (send and receive), to model the tanks and the operation sequence needed for each of them. Channeling constraints are formulated to govern pipeline sequencing and tank sequencing. A backtrack mechanism provides the foundation to the strategy for searching a solution. First, pipelines that are involved in a greater number of operations are chosen and delivery orders with the earliest due dates are sequenced first in these pipelines. Then, the volumes and tanks for undetermined delivery orders that were sequenced first in these pipelines are assigned. Finally, start and end times are assigned for all orders, which are again chosen by the earliest delivery deadline. 2 real instances are tested, both composed of 14 depots, 29 pipelines, 32 products, 242 tanks distributed among the depots. Instance 1 had a 10 day scheduling time horizon while instance 2 had a 7 day scheduling horizon. The primary goal was to search for a feasible solution.

Rejowski and Pinto [34] deal with the multiproduct pipeline scheduling problem. The system comprises the tank farm management at the refineries and at the depots, the pipeline operations and the required product demands at the local consumer markets. Scheduling applications address short term periods and deal with resource utilization such as tanks, pipelines and refinery production (from a few days to a few weeks). A MINLP formulation based on a continuous time representation for the scheduling of multiproduct pipeline systems that must supply multiple consumer markets is presented. Their formulation considers the booster stations yield rates with variable pumping costs. The MINLP presented is based on the MILP proposed originally in [30]. The mathematical formulation is presented and explained in detail. Hydraulic considerations and pumping yield rates are also considered. The proposed MINLP model achieved better results than those of the previously developed MILP that considers fixed flow and yield rate operations [31]. It is shown that the latter is a special case of the former. The examples presented have the same size of the previously introduced in [29, 30, 31]. A parallel between the continuous and discrete time representation is provided.

García-Sánchez et al. [10] present a methodology for addressing a real-world multi commodity pipeline scheduling problem. The problem addressed consists in obtaining a so called satisfactory schedule, which is a schedule with all its criteria being equal or better than certain satisfactory values for those criteria. The objective is to obtain a set of satisfactory schedules for the set of pumping stations based on six defined criteria. Each schedule consists of a series of K packages of different products where the k^{th} package of pumping station s referred as $PCK(k,s)$ is defined by the type of product $P_{pck}(k,s)$, the volume $V_{pck}(k,s)$, the level of flow rate enhancer injection $E_{pck}(k,s)$ which can be high, low or none; the splitting downstream along the different terminals. A *Tabu Search* implementation along with a simulation model of the system is introduced. The simulation model allows an accurate and suitable assessment of every particular schedule, whereas the Tabu Search guides the searching process and eventually succeeds in obtaining satisfactory schedules in terms of a set of relevant criteria. The objective of this work is to provide schedulers with useful tools to assist in their task. The problem addressed in this paper refers to systems that consist of a series of elements connected through pipelines of different length and radius. Such elements can be a refinery, a terminal or set of tanks, a branching node where 2 or more pipes are fed, and a branching terminal which is a terminal that feeds two or more pipes downstream. There is no reverse flow, multiple sources, the flow rate can either be constant or depend on the contents and splitting of the packages present in branching. The problem is defined by specifying the number of nodes N, the number of products P, the relation between nodes, its type (refinery, terminal or branching), the number and the identifiers of its immediate downstream successors. Also, initial contents, scheduling horizon, storage capacity and initial level of inventory, demands, delivery plan, available flow rates. The objective considered in this work is to obtain a set of satisfactory schedules for the set of pumping stations. Six criteria are used to define how good a schedule is: the amount of shortages, the times forbidden interfaces occur, the time during which some interface is stuck in a pipe or interface stoppages, the time during which a package cannot be pumped into a tank at the desired flow because it is full, or blockages; the cost associated with interfaces and the amount of volume non delivered.

MirHassani and Ghrobanalizadeh [23] present an integer programming approach to oil derivative transportation scheduling. A model for the pipeline transportation of petroleum products is presented. Pipelines connect refineries to local distribution centers where the products are sent through the pipe to satisfy the needs of consumer markets. The system reported is composed of an oil refinery, one multi-branch multi product pipeline connected to several depots and also local consumer markets which receive large amounts of refinery products. The pipeline is considered as a set of segments with equal volumes that connect the refinery to different depots. A path is a set of successive segments located between the refinery and a specific depot. The aim of the objective function is to account for the number of interfaces. They seek to arrange a pumping schedule with a minimum number of interfaces. The pipeline is considered as a set of segments with equal volumes that connect the refinery to different depots. They define a path as a set of successive segments located between the refinery and a specific depot. All the variables of the model are binary variables and most of them are defined by complicating constraints which makes the model intractable for bigger instances of the problem.

Cafaro and Cerdá [5] present a MILP multi period continuous time formulation for the so called dynamic pipelines scheduling problem (DPSP) suitable to handle multi period time horizons and considering multiple due dates for the product shipments. In this variant of the problem, pipeline operations are scheduled over a fixed length multi period rolling horizon. The pipeline schedule should be viewed as a dynamic timetable rather than a static one where only the scheduling decisions for the first or current period of the rolling horizon need to be implemented immediately. In contrast to the usual practice in the oil pipeline industry, the proposed approach accounts for nominated shipments with different promised dates, always occurring at period ends. Based on the new problem data, pipeline operations are optimally rescheduled through solving the proposed DPSP model. The results provided by the DPSP include an updated sequence and timing of the pumping runs inserting new batches in the pipeline over the current multi period rolling horizon, the product deliveries to distribution terminals taking place while executing a batch injection, the location and size of every batch inside the pipeline immediately before and after a pumping run, the updated projected inventories in refinery and depot tanks immediately before and after every new batch injection. This model can be extended to schedule pipeline networks with multiple exits not only for delivery of products to depot tankage but also for interchanging shipments with other outgoing pipelines at common terminals. The problem goal is to dynamically update the sequence and volumes of new product batches to be pumped in the pipeline throughout a multi period rolling horizon in order to meet every product demand at each terminal in a timely fashion, maintain the inventory level in refinery and terminal tankage within the permissible ranges, trace the size and location of every batch in pipeline transit and minimize the sum of pumping transition down time backorder and inventory carrying costs. The pipeline schedule should indicate the amount and type of product to be pumped, the batch pumping rate as well as the starting and completion time of every batch injection. 4 major sets define the mathematical formulation for the DPSP: the old (those already in transit along the line) and new (planned to be pumped in the pipeline at future periods) fungible batches, the pipeline distribution terminals, the refined petroleum products to be delivered from the refinery to terminals along the line and the time periods taking part of the multi period rolling horizon. Batch defining constraints are formulated to define the allocated product, initial batch size, initial injection time, final injection time, pumping run duration, completion time period. The batch dynamic properties are dependent of pipeline activity and their values change along the rolling horizon whenever a new batch is injected in the line. In order to control when a batch will arrive to a stated destination and what amount of product is to be diverted, the batch movement along the pipeline and the stripping operations to be executed while injecting a new product should be established. The problem constraints that are aimed to tracing batches and defining stripping operations are called batch tracing constraints. Batch tracing constraints involve a single set of binary variables through which the model can establish whether diverting a given batch to a certain depot while pumping a new batch or if it is not a feasible action. The entire line must be stopped if there is insufficient storage capacity at some depot to receive the specified amount of product from a batch in transit. A pipeline scheduling model should be capable of monitoring depot inventory levels to prevent from defining batch stripping operations causing tank overloading and product shipments from depots to neighboring markets that cannot be afforded due to lack of inventory. Depot inventory management constraints deal with issues related with demand satisfaction to minimize backorder costs. Refinery inventory management constraints are included to monitor the product inventories at the refinery. The algorithm for the periodic update of the pipeline operation comprises five stages: initialization, problem data update, pipeline schedule update, batch dispatching and horizon rolling and new instance generation. In the initialization stage, the DPSP parameters are set by the scheduler. The data updating stage, updates the input data for the current horizon. The core step of the algorithm is the pipeline rescheduling stage. It provides master planning over the current rolling horizon by running the multiproduct pipeline scheduling optimization system. The dispatching stage should account for the set of batch injections and batch stripping operations to be carried out between two consecutive points in time. Two instances are run for a proposed case study, a modified version of the single period real world case study introduced by Rejowski and Pinto [30]. The proposed formulation for the Dynamic Pipeline Scheduling Problem (DPSP) allows considering multiple due dates at periods ends.

Cafaro and Cerdá [6] introduce a new mixed integer linear programming formulation for the planning and scheduling of oil products pipelines operating on either fungible or segregated mode and featuring multiple input and output terminals. A continuous volume and time domain representation is used by this approach. A multi-source pipeline transports batches of oil products from various sources to many destinations. The complicating important features of multisource trunk pipelines, not present in the case of a single source system, are discussed. There are five important sets in the problem: pumping runs, batches, oil derivatives, oil refinery sources or input nodes and output terminals. Three different sets of binary variables are to be incorporated in the problem formulation to stand for the allocation of the oil refined products to batches, the assignment of batches and input nodes to the pumping runs, the destinations that receive some amount of product from the existent batch *i* during a given run. Also, continuous variables are considered such as the end time of any pumping run, the length of a given run, the size of the flowing batch i at the end time of any pumping run, the volume of the new batch I injected in the line from a given source during a given run, the upper coordinate of an existing batch at the end of a given run, the amount of product diverted from a given batch to a given output terminal during a pumping run. The model formulation comprises four blocks of equations related with: pumping run constraints, batch tracking constraints, feasibility constraints and product inventories in depot and tanks constraints. The problem goal is to minimize the total pipeline operating cost including the cost of underutilizing pipeline transportation capacity, transition costs, pumping costs and backorder costs. Two examples are solved using their model, one of them on a segregated mode and the other one on a fungible mode.

Relvas et al. [37] study a system comprising one pipeline that connects one refinery to one distribution center. It is desired to obtain the optimal pipeline schedule, with sequence of products, batches volumes, pumping rates and pumping and discharging timings; the inventory management at the distribution center, including daily volume balances by product and monitoring of arrivals, settling and approving tasks as well as satisfaction of clients demands. The objective function can be either economical or operational. They use an operational objective. The proposed architecture enables the interaction of the MILP model with the proposed model extensions: the reactive scheduling procedure and the sequencing heuristic. It also illustrates the connections that guarantee the feeding of input data and output results. There are three sources of inputs: initial conditions, market forecasts and scenario parameters. The MILP is run using inputs and possible heuristic results. The MILP model was built considering continuous representations of both time and pipeline volume. The demands are represented over a continuous scale. The products sequence may consist either on a periodic repetition of products or a free combination of products. The objective function used by the MILP model deals with operational indicators: total pumping time, total pumped volume, total final inventory, balance between total inputs and outputs, lowest final inventory among all products. The motivation for the heuristic presented in this work lies in the fact that decision makers and schedulers usually seek good solutions, close to the optimal, rather than a time consuming optimal solution with little margin of improvement when compared to the others; and the increasing use of decomposition approaches to reduce problem complexity. The system in study comprises one pipeline that connects one refinery to one distribution center. The refinery produces several oil products and the distribution center is responsible for supplying these products to a local market. It is desired to obtain the optimal pipeline schedule, with sequence of products, batches' volumes, pumping rates an pumping and discharging timings; the inventory management at the distribution center, including daily volume balances by product and monitoring of arrivals settling and approving tasks as well as satisfaction of clients' demand; while satisfying an operational objective function. The proposed architecture enables the interaction of the MILP model with the proposed model extensions: the reactive scheduling procedure and the sequencing heuristic. The inputs come from initial conditions, market forecasts and scenario parameters. The subsequent step decides what the type of sequence of products to be used is. The heuristic uses the initial state and market forecasts to analyze the current operational conditions and establish parameters to develop sequences of products. Fixed sequences are more suitable for larger time horizons while free sequences are easier to use in short time horizons. The MILP model is run using inputs and possible heuristic results. Model results are used as schedules for the current time horizon. The considered MILP model is the same as in [35, 36]. The used objective function considers the following operational indicators: total pumping time, total pumped volume, total final inventory, balance between total inputs and outputs, lowest final inventory among all products. A heuristic for resource allocation is proposed. This heuristic enables a study of the initial conditions combined with forecasted demands in order to develop suitable sequences of products for the given scenario. Once a set of sequences are obtained, they can be used to run the MILP model previously presented by Relvas et al. [35]. The heuristic procedure determines how the sequence should start, how to develop it along the time horizon and how it should end by developing an initialization for the sequence of products and giving margins for the maximum and minimum number of batches to be pumped. It also develops a set of the most adequate fixed sequences and run the model for all options. The initial level of inventory for the products is used to develop priorities. Higher priorities are given to those products that face stock out in first place if the pipeline would remain stopped along the time horizon. The bounds on the number of batches can be either based on products or based on cycles. The use of the most common batch volume is proposed by first obtaining the volume to be transported for each product necessary to cover the total demands. In this step, the product that is already inside the pipeline is considered. In the next step, the maximum and minimum number of batches in the sequence is obtained, after the products that are already inside the pipeline. Finally, the number of batches to be transported is determined. The values obtained in the previous step can now be used as new constraints in the model representation. A second method to determine the bounds on the number of batches was proposed that uses information based on a cycle of products. The input information concerns the total volume that is predicted to be delivered to customers. This should be the goal for the volume transported by the pipeline. The subsequent step is to analyze if there is the chance to establish a cycle unit of products. A cycle unit should cover every product to be transported within the system. The adopted strategy use some rules that consider the priorities and the bounds on number of batches: use as starting point the initial batches that are already inside of the pipeline, transport the products based on their priorities; after initialization, find a suitable point to start repeating cycles of products and terminate the sequence either with a cycle or with necessary batches in order to meet a number of batches between the calculated interval or the desired sequence stopping criteria. The procedure is repeated until all the possibilities within the interval of possible number of batches are totally covered. Three case studies are presented: case study 1 is used to illustrate and validate the heuristic procedure for a short term period of 1 week. Case study 2 is applied to a medium term horizon of one month and explores several options of the proposed heuristic. Case study 3 contemplates 6 consecutive months and the heuristic is applied consecutively providing the final data of each month to feed as initial conditions for the subsequent month. The results presented for each case study consider model performance and operational indicators as mentioned above. The proposed heuristic procedure combines scenario data and the matrix of possible sequences to derive valid sequence directions in order to improve the solution method. Two strategies were proposed. The first one was applied without success to a medium term scheduling problem. The second strategy provides good results in both, short term and medium term scheduling horizons. The approach presented aims to provide decision support for the user as well as to reduce model complexity.

Research Perspectives in this Field

In this section, perspectives of the research on the problem of transportation of refined products via pipeline systems are presented, conclusions of these perspectives are provided and fertile research directions for this problem are proposed. In section 3 of Rejowski and Pinto [33] the most significant literature related to pipeline operations is displayed in a table by author, problem (product transportation/crude oil transportation), operation (scheduling/planning), type of formulation (mixed integer and linear programming, linear programming, object oriented tools and artificial intelligence), time representation (either if it is unavailable or available, in which case it can be either discrete or continuous) and the solution approach presented (single model or decomposition strategy). The most up to date summary of the state of the art in pipeline transportation research is presented in the aforementioned paper. In spite of the significant number of mathematical approaches, in no case reported in this table there is a network flow based formulation where delivery time windows are considered and a decomposition approach implemented to solve it. In order to broaden the view of the literature surveyed for the problem of interest in this dissertation, 6 tables are presented.

Instance notation. As part of the contribution of this dissertation, a notation for the instances of this problem to characterize the main aspects of their topology is first introduced in this section. Such a notation was not available in the relevant literature for this problem until now. In the surveyed papers, different types of instances of the problem were found having different system features. For this reason, it is important to have a common notation to characterize the different scenarios of the problem found in the most representative publications in the literature of this field. The proposed notation considers the following features:

i. The number of sources, represented by S
ii. The number of destinations, represented by D
iii. The number of products, represented by P and
iv. The type of flow, represented by F. Flow can be either reversible or not reversible. The vowel u will indicate that flow travels in a Unidirectional way and the letter b in the corresponding field will be employed when bidirectional flows can occur
v. The type of pipeline configuration, which can be either *single pipeline*, represented by s, or *network of pipelines*, represented by n.

The proposed notation for the topology of a family of instances of this problem is:

$$S/D/P/F/\pi$$

where S, D and P are integer numbers. The parameter F can be either u or b depending on whether the flow in the line is unidirectional or reverse flow operations are present. The parameter π can either be s or n depending on whether the instance has a single pipeline or a network of pipelines. The information included in the proposed notation perhaps include the most important features of any instance of the problem under consideration that can be found on the literature and allows a classification to be made for the different situations considered in the literature, aiming to provide a better understanding of the state of the art in pipeline transportation systems for multiple refined products.

Table 1, Table 2 and Table 3 present a taxonomy of the surveyed papers by modeling features, optimization criteria and solution strategy. Table 4, Table 5 and Table 6 report the different topologies of the instances of this problem considered in previous research efforts.

Three types of modeling methodologies can be distinguished from Table 1, Table 2 and Table 3: Mathematical/MILP related modeling approaches, simulation modeling approaches, and heuristics and artificial intelligence based approaches. The most important contributions in MILP approaches are made by Hane and Rattliff, Cafaro and Cerdá, Magatão et al. and Rejowski and Pinto. Almost 50% of the research efforts were MILP related implementations. Almost two thirds of the mathematical approaches considered a single problem type of solution procedure and only one third proposes a decomposition scheme related procedure to solve the model. Regarding the objective function, about two thirds of the MILP related publications considered optimization of costs and about one third focuses the attention in operational objectives. It is important to remark that about one third of the publications related with mathematical approaches either does not consider the costs of the interfaces or the information is not specified. Among all the publications, this proportion increases to two out of every five publications. Only one third of the MILP implementations rely on a continuous modeling approach while the remaining portion of surveyed papers relies on a discrete modeling approach. About 70% of the surveyed publications were mathematical modeling and solution based approaches. The other papers used simulation, heuristics and in a smaller proportion artificial intelligence.

Probably the classic instance in the literature for this problem was first introduced by Rejowski and Pinto [29]. This instance is also used in Rejowski and Pinto [30, 31, 32 and 33], and also in Cafaro and Cerdá [3, 4 and 5]. The scenario consists of one refinery, five depots, four products and a single pipeline with unidirectional flow. The use of this instance is reported in about one third of the publications found in the literature of this problem to test the different modeling approaches and solve the problem seeking the optimal solution and not just a feasible pumping sequence.

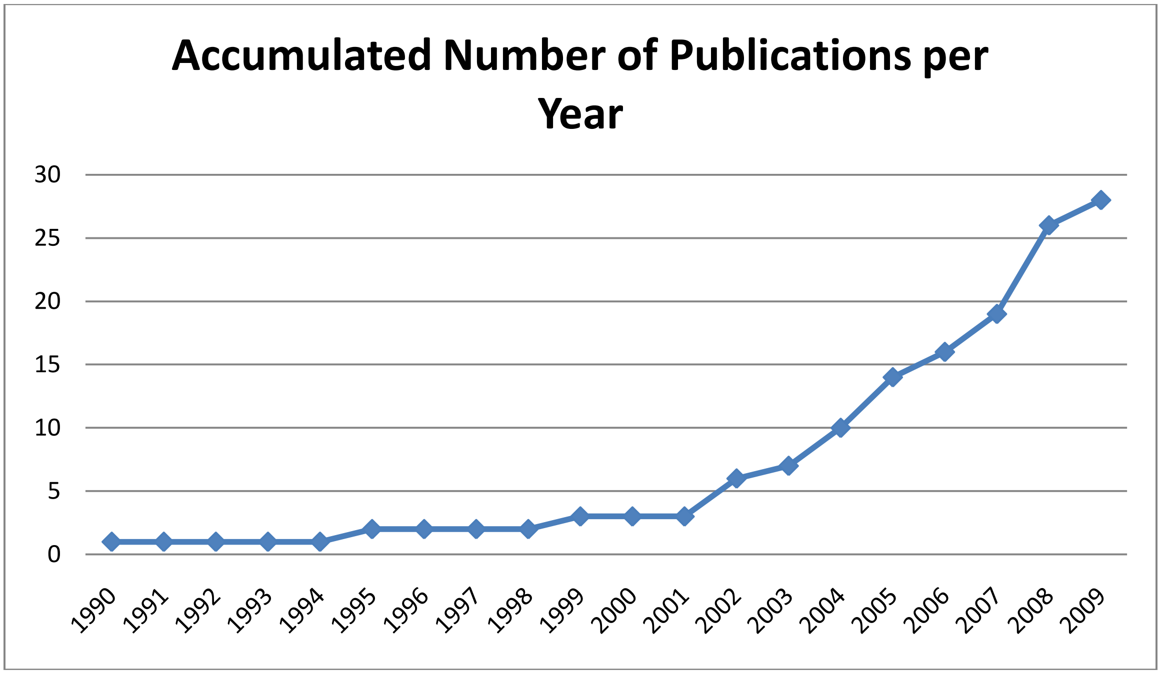

Figure 1: Accumulated Number of Publications at the End of Each Year from 1990 to 2009.

Figure 1 suggests that this problem has an increasing importance. The first paper directly related to this problem was published by Camacho et al. [7] in 1990 and along with the contribution of Hane and Ratliff [15] in 1995; both remained for almost 10 years as the only visible published attempts to provide a scientific approach to deal with this economically important problem. After the year 2000, the problem has been gaining more attention from the operations research community and several publications have been released in the past 10 years. Perhaps, the majority of the work on this problem has been published in the last 10 years. One might say that this is a relatively new problem enjoying increasing attention from the community.

Table 1: A Taxonomy of the Literature: by Modeling Features, Optimization Criteria and Solution Strategy: Book Chapters

Reference	*Type of Model*	*Time/Volume Approach*	*Optimization Criteria*	*Accounts for Interfaces*	*Solution Strategy*
Milidiú et al. [21]	AI	Discrete	Operational Objectives	Yes	AI
Garcia-Sanchez et al. [10]	Simulation	Continuous	Interface Costs & Operational Objectives	Yes	TS
Moura et al. [24]	Constrained Programming with nonlinearities	Continuous	NS	No	Construction heuristic & CP

Table 2: A Taxonomy of the Literature: by Modeling Features, Optimization Criteria and Solution Strategy: Conference Papers

Reference	*Type of Model*	*Time/Volume Approach*	*Optimization Criteria*	*Accounts for Interfaces*	*Solution Strategy*
Camacho et al. [7]	Simulation	Continuous	Power Costs	No	Simulation
Sasikumar et al. [39]	Knowledge-Based Approach	Continuous	Schedule Cost & Operational Objectives	Yes	Heuristic search
Crane et al. [8]	Chromosome	NA	Operational Objectives	No	GA
Rejowski and Pinto [29]	MILP	Discrete	Inventory, Pumping and Transition Costs	Yes	Single Model
Milidiú et al. [22]	AI	Discrete	Operational Objectives	Yes	AI
Cafaro and Cerdá [4]	MILP	Continuous	Inventory, Pumping and Transition Costs	Yes	Decomposition
Magatão et al. [19]	CLP & MILP	Discrete	Inventory, Pumping and Transition costs	Yes	Decomposition
Rejowski and Pinto [32]	MINLP	Continuous	Inventory, Pumping and Transition costs	Yes	Single Model
Relvas et al. [34]	MILP	Continuous	Operational Objectives	No	Single Model
Maruyama Mori et al. [20]	Simulation	Continuous	NA	No	NA/NS
Neves-Jr. et al [27]	MILP	Continuous	Operational Cost Factors	NA/NS	Decomposition
Moura et al. [25]	Heuristic and CP	NS	NS	NA	Decomposition
Neves Boschetto et al. [26]	MILP	Continuous	NA	Yes	Decomposition

Table 3: A Taxonomy of the Literature: by Modeling Features, Optimization Criteria and Solution Strategy: Journal Articles

Reference	***Type of Model***	***Time/Volume Approach***	***Optimization Criteria***	***Accounts for Interfaces***	***Solution Strategy***
Hane and Ratliff [15]	MILP	Discrete	Power Costs and Operational Objectives	No	Decomposition
Rejowski and Pinto [30]	MILP	Discrete	Inventory, Pumping and Transition Costs	Yes	Single Model
Rejowski and Pinto [31]	MILP	Discrete	Inventory, Pumping and Transition Costs	Yes	Single Model
Cafaro and Cerdá [3]	MILP	Continuous	Inventory, Pumping and Transition Costs	Yes	Single Model
Magatão et al. [18]	MILP	Discrete	Inventory, Pumping and Transition costs	Yes	Decomposition
De La Cruz et al. [9]	MOEA/MILP	Discrete	Operational Objectives	Yes	2 Single Models & a Hybrid
Relvas et al. [35]	MILP	Continuous	Operational Objectives	No	Single Model
Relvas et al. [36]	MILP	Continuous	Operational Objectives		
MirHassani and Ghorbanalizadeh [23]	MILP	Discrete	Operational Objectives	Yes	Single Model
Rejowski and Pinto [33]	MINLP	Continuous	Inventory, Pumping and Transition Costs	Yes	Single Model
Cafaro and Cerdá [5]	MILP	Continuous	Cost of underutilizing pipeline transportation capacity, transition costs, pumping costs and backordering costs	Yes	Single Model

GA: Genetic Algorithms, AI: Artificial Intelligence, NA/NS: Not Available or Not Specified

Table 4: A Taxonomy of the RPDPPS Literature: by Instance Main Features: Book Chapters

Reference	$S/D/P/F/\pi$	***Pipeline Length/Volume***	***Company/Country***
Milidiú et al. [22]	NS/NS/NS/b/n	NS	Random
Garcia-Sanchez et al. [10]	1/4/7/u/n	18490 m^3	Compañía Logística de Hidrocarburos/Spain
Moura et al. [24]	4/NA/NA/b/n	NA	Petrobras/Brazil

Table 5: A Taxonomy of the RPDPPS Literature: by Instance Main Features: Conference Papers

Reference	$S/D/P/F/\pi$	***Pipeline Length/Volume***	***Company/Country***
Camacho et al. [7]	1/4/5/u/s	NA	CAMPSA/Spain
Sasikumar et al. [39]	1/3/4/u/s	500 Km	Indian Oil Corporation/India
Crane et al. [8]	1/7/2/u/n	NA	Williams Energy Group/USA
Rejowski and Pinto [29]	1/5/4/u/s	475 m^3	REPLAN Refinery (Petrobras)/Brazil
Milidiú et al. [22]	1/4/NS/b/n	NA	Random
Cafaro and Cerdá [4]	1/5/4/u/s	475 m^3	REPLAN Refinery (Petrobras)/Brazil
Magatão et al. [19]	1/1/4/b/s	7314 m^3, 97.5 Km.	Random
Rejowski and Pinto [32]	1/5/4/u/s	475 m^3	REPLAN Refinery (Petrobras)/Brazil
Relvas et al. [34]	1/1/6/u/s	NA/NS	Companhia Logística de Combustiveis (CLS)/Portugal
Maruyama Mori et al. [20]	1/6/8/b/n	42000 m^{3*}	Random
Neves-Jr. et al. [27]	3/5/10+/b/n	NA/NS	Petrobras/Brazil
Moura et al. [25]	1+/20+/10+/u/n	NA/NS	Petrobras/Brazil
Neves Boschetto et al. [26]	NA/NA/NA//NA	NA	Random

vu: Volumetric units (not specified); *: Information available for only 1 of 15 pipes

Table 6: A Taxonomy of the RPDPPS Literature: by Instance Main Features: Journal Articles

Reference	$S/D/P/F/\pi$	***Pipeline Length/Volume***	***Company/Country***
Hane and Ratliff [15]	1/5-30/NA/u/s	2885 Miles	Colonial Pipeline/USA
Rejowski and Pinto [30]	1/5/4/u/s	475 m^3	REPLAN Refinery (Petrobras)/Brazil
Rejowski and Pinto [31]	1/5/4/u/s	475 m^3	REPLAN Refinery (Petrobras)/Brazil
Cafaro and Cerdá [31]	1/5/4/u/s	475 m^3	REPLAN Refinery (Petrobras)/Brazil
Magatão et al. [18]	1/1/4/b/s	7314 m^3, 97.5 Km.	Random
De La Cruz et al. [9]	2/3/4/b/n	NA/NS	Random
Relvas et al. [35]	1/1/6/u/s	18000 vu147 Km.	Companhia Logística de Combustiveis (CLS)/Portugal
Relvas et al. [36]	1/1/6/u/s	18000 vu147 Km.	Companhia Logística de Combustiveis (CLS)/Portugal
MirHassani and Ghorbanalizadeh [23]	2/3/3-4/u/n	319 Km	Random
Rejowski and Pinto [33]	1/5/4/u/s	163000 m^3	REPLAN Refinery (Petrobras)/Brazil
Cafaro and Cerdá [5]	2/3/3/u/s	80 vu	Random

The real instances reported in the literature for this problem are representations of systems of distribution for petroleum products via *polyducts* in Brazil, Portugal, India and USA, where the highest number of research efforts are made in Brazil and the lowest in the United States. One of the instances that is reported more frequently in the surveyed papers corresponds to a pipeline system in Brazil where 4 products are to be delivered to 5 depots from one single refinery through a unidirectional flow pipeline with a volume capacity of 475 cubic meters. However, in Rejowski and Pinto [33] the same system is reported with a different volume. This time, a 163000 cubic meters pipeline is used to distribute the same 4 products to the same set of depots. Only 5 out of 27 publications addressed the problem for the case of multiple sources: Moura et al. [24], Neves-Jr. et al. [27], De La Cruz et al. [9] and MirHassani and Ghorbanalizadeh [23]. When the topologies are small, authors focus the attention in the optimization of either operational factors or cost related objective functions. Also, the level of detail of the system is considerably higher than those achieved in papers where larger topologies are considered. Hane and Ratliff [15] are some of the pioneers of this problem. The level of detail at which they addressed the problem is different than the level at which other important contributors have attempted to solve it, such as Rejowski and Pinto, Cafaro and Cerdá, Magatão et al., and others. It is important to remark that for the largest instance considered in the literature at the highest level of detail, the objective of the problem was to provide a feasible solution and no optimization process was considered.

Conclusions

The problem of product distribution in the petroleum industry has been addressed from different perspectives using different techniques and also, for different decision levels. Perhaps, the pipeline transportation problem of multiple refined products from a refinery to several market zones is at an operational decision level indeed.

The refined products distribution problem via pipeline systems (*RPDPPS*) has been gaining more attention in the past 10 years from the operations research community. Despite the broad applications of management science in the petroleum industry, probably the first visible attempt to solve this specific problem was published only 18 years ago and remained like that for almost 10 years. The problem has been addressed in a broad range of real life scenarios with different sizes, objectives and in various countries as well being Brazil, Portugal, USA, Spain and India those where most of the contributions for this problem come from, as shown in Figure 2. Despite the real life applicability of this problem, the instances reported in an important portion of the surveyed relevant publications are actually random and do not correspond with a reported real life system.

Mathematical, simulation, heuristic and artificial intelligence approaches have been employed to provide either a feasible solutions, optimal or near optimal solutions for the problem.

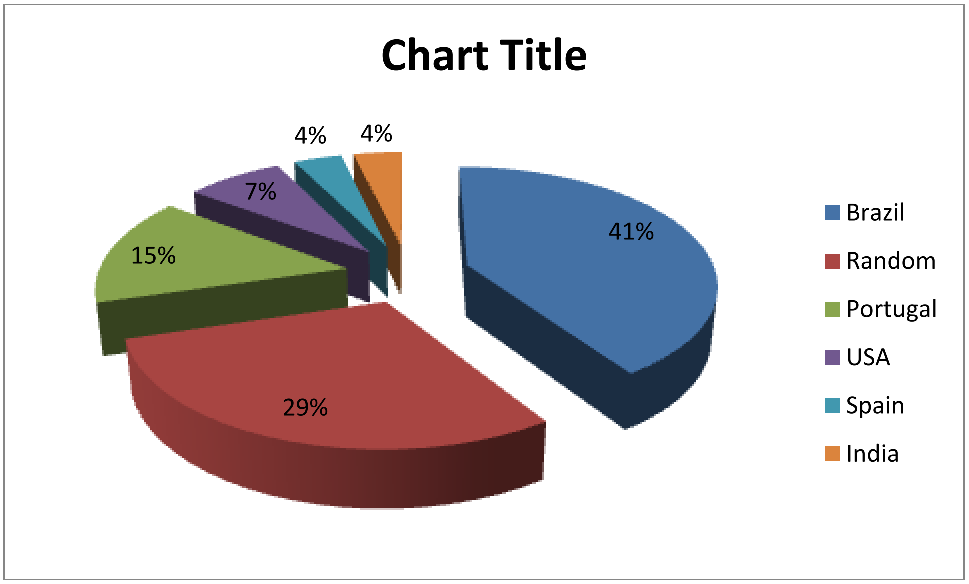

Figure 2: Countries reporting research contributions for the RPSDPPS

Decomposition approaches have not been explored intensively in the previous attempts to solve this problem. Compared with the real life cases in the United States, the instances studied in the available literature seem to be small. No network flow based modeling approaches at an operational level were reported in the surveyed papers. Meta heuristic techniques such as *simulated annealing*, *genetic algorithms* and *tabu search* have not been fully explored in implementations to solve this problem. Perhaps, the most promising approaches involve the use of all these techniques in decomposition based solution procedures. The creation of an advanced basis with the use of meta heuristic techniques to help guiding the optimization procedure in its early stages and the use of mathematical procedures to improve this initial point is probably a fertile strategy that has not been explored for this problem either.

Another important fact is that the problem structure was not considered and exploited in any of the mathematical formulations proposed. *Brute force* branch and bound procedures were used to solve the problem with neither the use of decomposition schemes such as Branch and Price nor with the implementation of branching rules and priorities, in the case of branch and bound implementations. The use of column generation techniques has not been reported in the most representative papers surveyed where mathematical approaches were proposed.

New research about this problem should be directed to address these gaps. This is a relatively new problem and there are still modeling and solution approaches that have not been explored. Also, the inclusion of additional features of the real life system in the model, such as changes in flow rates, leaks, maintenance stoppages and others, is to be considered in future modeling attempts. On the other hand, simulation modeling approaches have been used to represent the operational characteristics of the systems with more accuracy but only in rare cases, in presence of an optimization tool. Simulation optimization approaches seem to be a promising research path on this field.

In this dissertation, a novel network flow modeling approach for multiple commodities is introduced. This approach can be used to represent any of the topologies found in the literature. The problem structure is presented and based on it the convenience of its exploitation in a column generation procedure is explored. A Branch and Bound algorithm that exploits the dynamics of the system is proposed as well by defining branching rules and priorities based on the logic of the operational aspects of the system. The computational performance of such an approach is shown to be superior than both decomposition based approaches and brute force branch and bound solution methodologies. Computational experiments are reported for the decomposition based solution procedure, brute force branch and bound and the proposed branch and bound with branching rules and priorities.

CHAPTER III
OVERAL CONCEPTUAL APPROACH

Time representation is one of the most important issues in an optimization model [13]. The model introduced in this dissertation considers the occurrence of important events only at discrete points in time evenly separated each from one another. The time that separates 2 consecutive discrete points in the horizon planning period is the time it takes a batch of product to traverse the pipeline from its current position to the next one downstream at the rate of flow of the system, which is assumed to be constant. The scheduling horizon is divided into a finite number of time intervals of equal duration which makes this a discrete model. The transfer of product from the pipeline to a given depot happens during the time interval and is assumed to finish at the end of a given time interval. Although the flow in this system is continuous, the conservation of flow constraints in the pipeline are monitored only at these finite discrete points in time where the important changes in the system might happen, e. g.: a batch of product completes its entrance at a given depot and a new batch of product is injected into the pipeline. The size of the model depends on the number of time intervals of the horizon planning period under consideration, which is at least the same as the total number of batches to be pumped through the pipeline, and the size of the pipeline, given in number of batches. In the case that no disruptive operation takes place during the horizon planning period, it is the same as the number of batches to be pumped through the pipeline. If that is the case, the product flow through the pipeline never stops. Discrete formulations have proven to be very efficient for a wide variety of industrial applications [13].

Modeling Paradigm

The pipeline is divided in segments and each segment is divided in batches. The proposed modeling methodology borrows this concept from Rejowski and Pinto [29] who reported its use by pipeline operators in the literature for the first time. The content of each batch of the pipeline may or may not change over time. Based on this idea, the proposed modeling approach is based on the representation, at discrete points in time, of the journey of the batches of product through the different batch positions inside the pipeline using a network flow model for multiple commodities and an additional set of constraints. Each pipeline segment has a known capacity in number of batches [29]. The journey of each batch of product starts at the moment it leaves the refinery and finishes when it reaches its destination at the corresponding depot in order to satisfy its demand. The modeling approach aims to represent the evolution of the pipeline during the entire horizon planning period by focusing on discrete points in time at which the state of the system might have changed instead of tracking the system continuously through the horizon planning period. This change depends on whether at any given point in time a certain batch continues its journey downstream or stops it because another batch of product is being received upstream the line.

The decisions taken in the system correspond with the following 2 questions:

1. What is the position of each batch in the optimal sequence of products to be pumped through the pipeline?
2. What is the depot or market zone at which each batch has to be pumped in order to satisfy its demand?

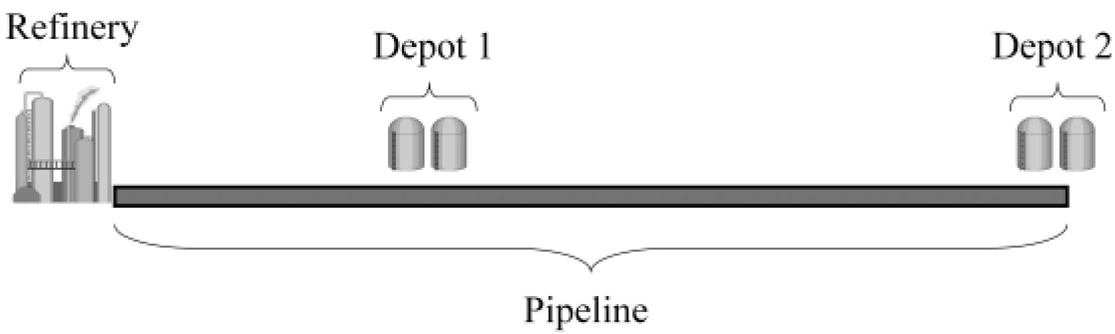

Figure 3: Representation for a Small Instance of the Problem

These two decisions will determine the following important outputs for the system:

- The position in the sequence for each batch of product
- The destination at which each batch of product is sent
- The position inside the pipeline for each batch of product during each point of the horizon planning period
- The Profile for the Inventory of each product at the refinery and at the depots

The model must describe the evolution of this system through the entire horizon planning period given the simplification assumptions made, at a finite set of discrete points in time. If a batch of product is injected into the pipeline from the refinery, then all the batches that are currently inside the pipeline move downstream the line and somewhere in the system, a batch of some product has to be accepted to release space inside the pipeline so then the new batch that is being injected into the pipeline can enter the line. Because this kind of pipelines is used to move different types of products, they are shipped in batches[4]. A batch is a standard measure for the minimum

[4] www.petrostrategies.org

transported that is used by pipeline companies as one of their tools to control their operation. The bigger the batch size is the better for the operation of the pipeline.

In order to better illustrate the proposed modeling concept, a small scenario for the problem is considered. In this instance of the problem, that will be called instance A; a refinery has to send 2 products towards 2 depots. Figure 3 provides a representation for this scenario.

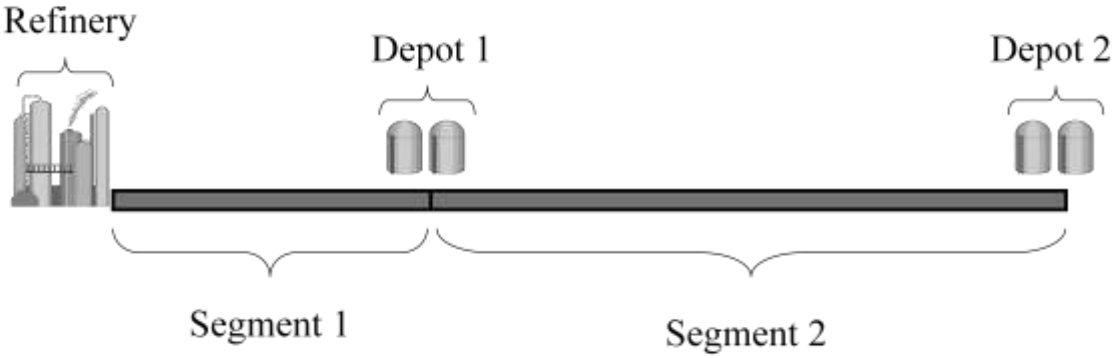

Figure 4: Pipeline Divided in 2 Segments

The pipeline is now divided in two segments, one from the refinery to depot 1 and another from depot 1 to depot 2. Figure 4 provides an illustration of this idea. In this small scenario, the first segment has a volume capacity to store 2 batches and the second segment has a volume capacity to store 4 batches. Figure 5 illustrate this.

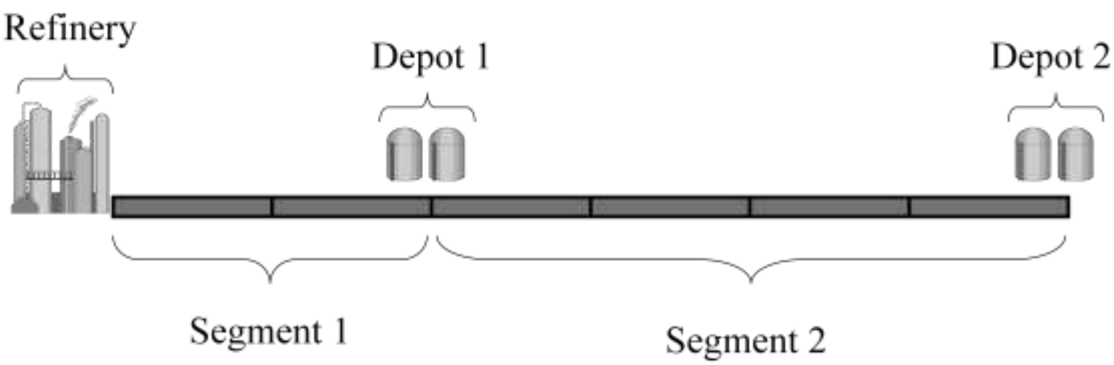

Figure 5: Pipeline Divided in Batches

The variation in the energy costs is one of the most important concerns in this problem since the pumping costs are determined by the energy costs [86]. At consumption peaks, it is more expensive to pump products downstream the line. If the pipeline is full with one single product and the customer requirements allow it, it is cheaper to stop the flow where a peak in energy prices is happening than to continue the flow in the system. In a situation like this, the system remains in the same state from a given point in time *t* to the next point in time *t+1*. This scenario is represented in Figure 6.

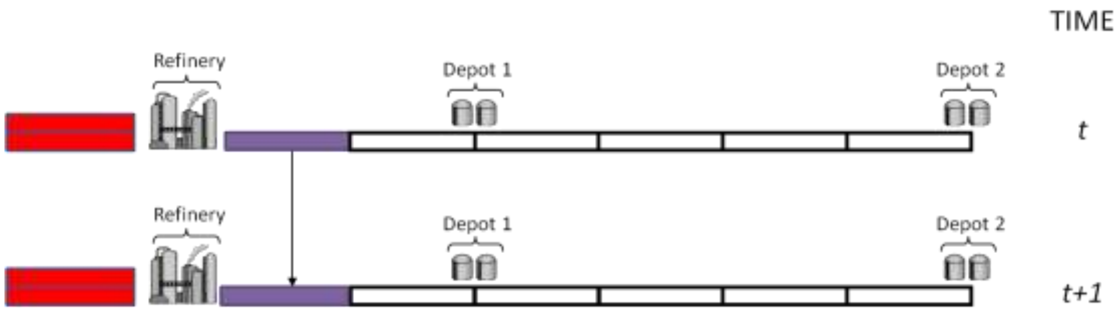

Figure 6: No Product is injected into the Pipeline at Time t

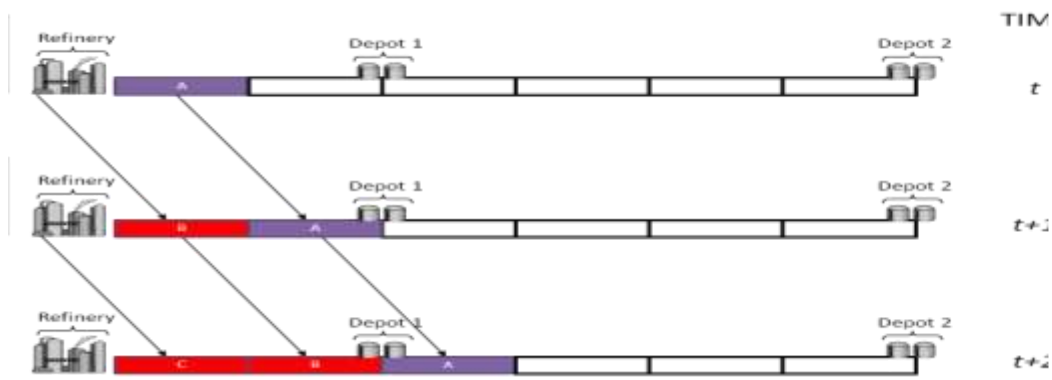

Figure 7: 2 Batches of Product are injected into the Pipeline at times t and t+1

In case that the 2 batches of product available at the refinery at time t are pumped into the pipeline, another scenario for the different states of the system can be as represented in

Figure **7.** In this case, the batch of product A is pushed downstream the line when the batches of product B and C are injected into the line at time t and t+1. The product that was into the pipeline at time t is not received at depot 1 at time t+2 so then it continues its journey downstream the line and occupies the first batch position of segment 2 of the pipeline at time t+2.

Now, if batch of product A is received at depot 1 at time t+2 instead, another scenario is observed that is also included in the representation of the system by the proposed modeling methodology.

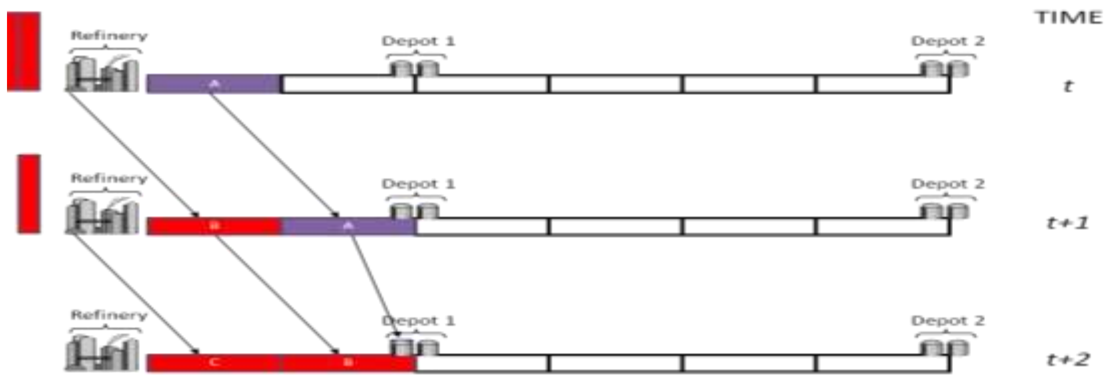

Figure 8: Product A is Received at Depot 1

As shown in Figure 8, a batch of product *A* is received at Depot 1 at time t+2 in order to satisfy its demand and whatever is downstream the line inside segment 2 keeps its position from time t+1 to time t+2. These dynamic aspects of the system have to be very well represented by the proposed methodology since they constitute inherent characteristics of this type of system.

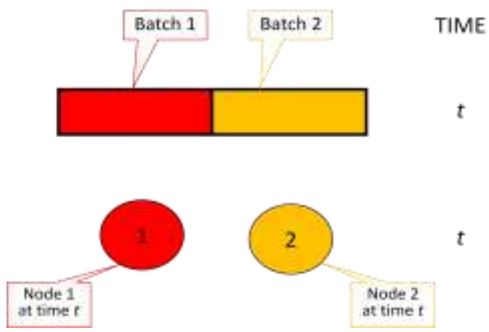

Figure 9: First Segment of the Pipeline Containing 2 Different Products

These are all the important different possibilities that have to be considered to propose the modeling approach. Recall that the first segment of the pipeline has a volume capacity of 2 batches and the second segment has a volume capacity of 4 batches. Consider now the first segment of the pipeline. By representing each of the 2 batches using a node, segment 1 of the pipeline at any point in time t can be modeled by a set of nodes as shown in Figure 9, each one representing a batch position inside this portion of the pipeline. In addition to this, the representation for the depots can be made using an equivalent approach. A sink node can be used to represent the tank farms at each market zone. This node is an aggregate representation of the different tanks at the corresponding depots. Using this modeling idea, the representation of the pipeline and depots at any point in time is as shown in Figure 10

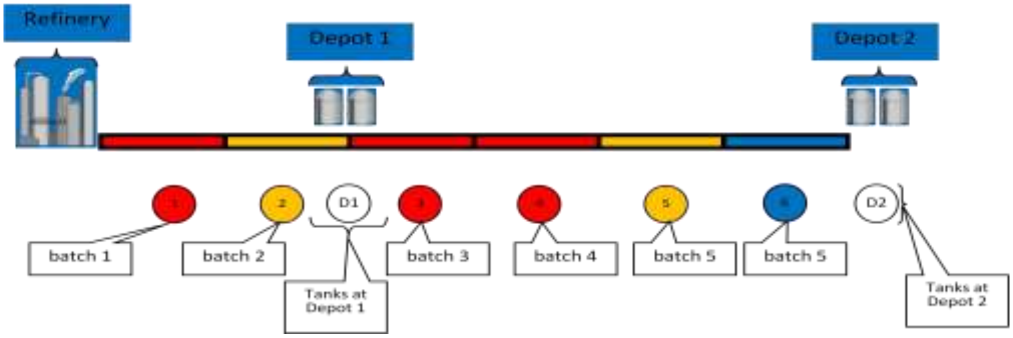

Figure 10: System Representation at any Point in Time *t*

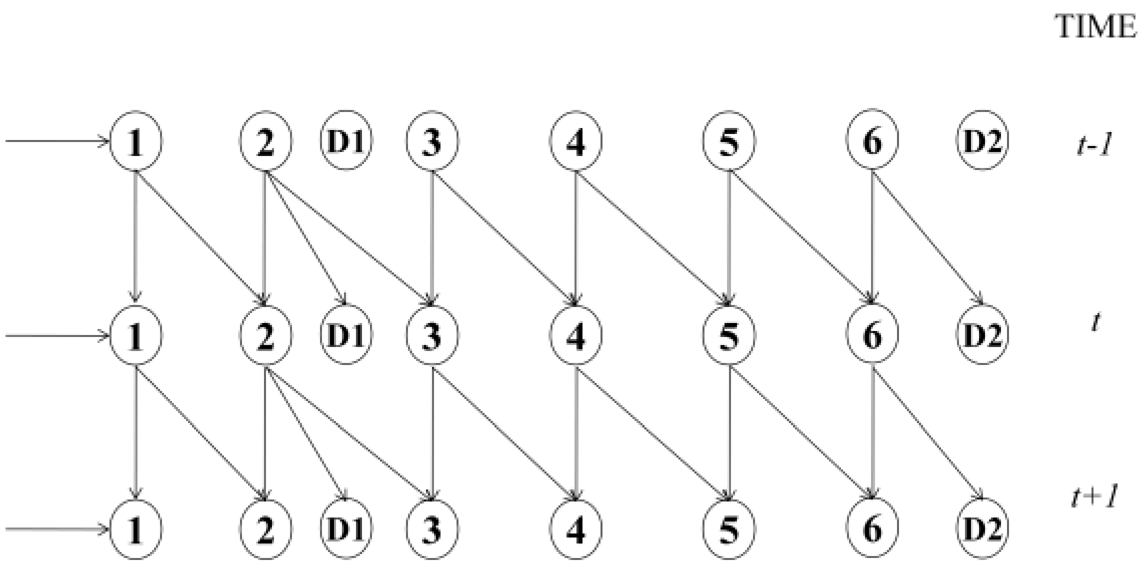

Figure 11: System Representation Between Times $t-1$ and $t+1$

Based on the possible scenarios represented in Figure 6,

Figure **7** and Figure 8 and the way the pipeline and depots are represented at any point in time t, the representation of the pipeline and the possible changes of its state between times $t-1$ and $t+1$ is as shown in Figure 11. The arcs entering node 1 represent the injection of any product from the refinery. This representation can be accomplished by defining a basic component of the network model composed by the nodes representing the products available at the refinery ready to be sent, the nodes corresponding to the batch positions inside the pipeline for the different segments in which it is divided, the nodes that represent the tank farms at the different market zones in an aggregate fashion and the arcs representing all the possible changes in the state of the system given by the movement of the product inside the line. Recall that the horizon planning period consists of a set of discrete points in time that is at least as big as the number of batches to be pumped from the refinery to the depots. Each point in time of the horizon planning period for any scenario of this problem can be represented using the basic component of the network.

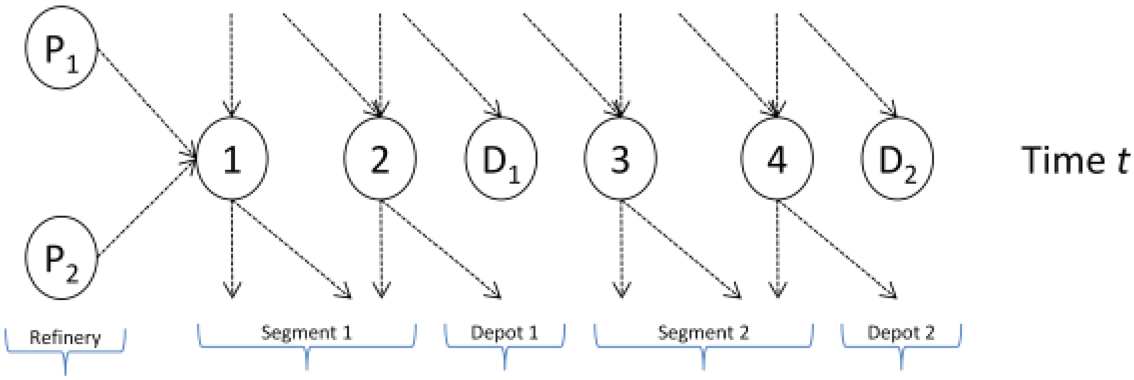

Figure 12: Basic Component of the Network of *Instance A*

The basic component of the network for this case is provided in Figure 12. The source nodes P_1 and P_2 correspond to the products 1 and 2 available at the refinery and ready to be sent through the pipeline in order to satisfy their demand at the market zones. The sink nodes D_1 and D_2 correspond to the depots downstream the line. The intermediate nodes 1, 2, 3 and 4 correspond to the batches in which the pipeline is divided for the analysis. For the general case where there are D depots, P products and the capacity, in number of batches, for each segment $l, l = 1,2,\cdots,D$ is given by CS_l. Figure 13 provides the basic component of the network for the general case.

Source Nodes, Intermediate Nodes and Sink Nodes of the Network Representation

There are two types of source nodes in the proposed network modeling approach. The first type is represented by the leftmost set of nodes modeling the products available at the refinery and ready to be sent through the pipeline. There is one node for each product and its capacity is equal to the number of batches of the corresponding product available at the refinery. The second type of origin nodes is represented by the nodes modeling the batches of the pipeline at the beginning of the horizon planning period. At time zero, the pipeline is containing determined products inside it. The nodes representing the batches of the pipeline at time zero are the second type of origin nodes. Each one has a capacity equal to one for the product that is currently inside the corresponding batch of the pipeline. The elements of the set of intermediate nodes are the nodes representing the pipeline from time 1 to time $T-1$, where T is the total number of discrete points in the horizon planning period. On the other hand, the sink nodes are classified in two types as well: the nodes representing the depots, at which the demanded products are received, and the nodes representing the pipeline at time T

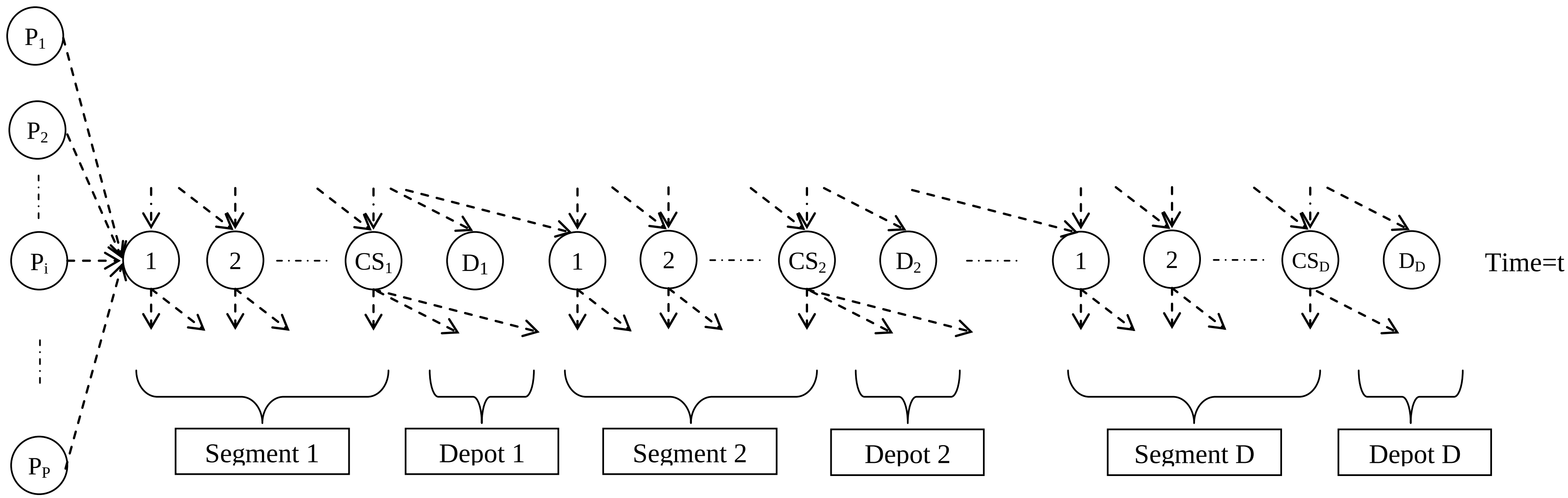

Figure 13: Basic Component of the Network for the General Case

The final state of the pipeline is a parameter of the problem. This new modeling concept can be used for system representation in presence of multiple sources and multiple destinations. Also, pipelines with branches can be modeled using this modeling approach. All the topologies of the system found in the literature so far can be modeled with this approach, considering the assumptions made.

Conclusion

A novel modeling approach has been introduced for the refined products distribution problem via pipeline systems. The proposed conceptual approach allows the representation of the different topologies of the system reported in the literature using a network model where a set of nodes represents the sources, another set of nodes represent the destinations and the pipelines are modeled by a corresponding set of nodes as well, each node modeling a corresponding batch position inside the pipeline. The dynamics of the system are also an important feature that has to be considered when modeling it. The proposed modeling concept can represent the events that may change the state of the system by creating flow between the nodes of the network through a selected set of arcs and by representing the system at discrete points in time using the basic component of the network introduced in Figure 13. Perhaps a suitable representation of the system dynamics through the horizon planning period is the most challenging part in the modeling phase to study this problem. The modeling concept proposed for this problem is another contribution of this dissertation.

CHAPTER IV

MATHEMATICAL MODEL

A modeling approach based on a network structure was introduced in the last chapter. The basic dynamic aspects of the system were shown to be captured by the network model through the horizon planning period. The interpretation of the nodes and arcs of the network and their role in the proposed model were explained and related with the real system. The basic component of the network, which is replicated through the horizon planning period as many times as discrete points it contains was also introduced. The network corresponds to a multi commodity network flow model with binary variables. In this chapter, the mathematical representation of the multi commodity network flow model as well as an additional set of constraints necessary to model the dynamics of the system commented in the previous chapter are presented. The problem structure is studied and model insights are provided.

Mathematical Model

This section provides the mathematical formulation for the network structure that represents the system of interest and its evolution through time introduced in the last chapter. Assumptions made to model the system are first commented and the optimization criteria presented, model parameters are provided, decision variables defined, the objective function is stated and the sets of constraints for the model displayed and explained in detail.

Assumptions

- The demand for each product as well as the capacity for each segment is given in the same units of volume –batches-, and are known as well as its availability in the refinery
- Pipeline segments are always used at full capacity and their capacities are known.
- All the terminals along a delivery line can accept shipment at the full rate of flow
- The products move through the pipeline at a known constant rate of flow given in batches per time unit
- The size of the sequence of all products to be pumped is measured in number of batches.
- There is no reverse flow in this system.
- Momentum propagates instantaneously through the pipe which implies that if an amount of product x enters the pipeline, another amount of product x must exit it.

Optimization Criteria

Dispatching petroleum products may involve the consideration of transportation and product sourcing costs, operating rules of the transportation units, inventory considerations, customer service policies, and other factors [38]. Different optimization criteria have been considered for this problem. Camacho et al. [1] considers the minimization of the energy costs. Hane and Ratliff [15] consider the minimization of a surrogate for pumping and maintenance costs. In [1, 15, 29, 30, 31 and 34] the minimization of the pumping costs, inventory holding costs at the refinery and transition costs, which is the cost of sequencing two non-miscible products consecutively, are considered to be optimized. In addition to this, [1] also considers as objectives to keep the pipeline running as close as possible to maximum capacity, enhance shipper information about the status of product movements and take advantage of time varying energy costs for pump power. Other operational objectives have been considered in previously published papers for this problem. Such objectives include:

- *Minimize the deviation from target values for shortages*
- *Minimize the non-delivered volume*
- *Minimize the number of forbidden interfaces/interface stoppages/blockages*
- *Minimize the deviation of the stock levels from target values during the horizon planning period*
- *Maximize the level of satisfaction*
- *Minimize the time in which demand is satisfied and product changes in the polyduct*
- *Maximize amount of product transported plus total inventory at the end of time horizon*
- *Minimize the difference between products transported and outputs to clients and Maximize the total pumping time*

In this dissertation, the performance of the system is defined as a composite cost function where pumping costs and inventory costs are considered. An extension where transition costs are considered is presented and computational experiments are provided for this problem variation. Figure 13 in the previous chapter provided the illustration of the basic component of the network approach that is used to represent the system. In this figure, the pipeline is composed of as many batches, which corresponding position is represented by the nodes, as its volume capacity. The model keeps track of the pipeline content evolution on a batch basis at discrete points in time. This role is performed by the intermediate nodes of the network. The sink nodes are the nodes representing the depots at which the products are received and also a special set of sink nodes is devoted to represent the final state of the pipeline.

Let us now consider the first node from left to right in the set of nodes corresponding to segment 1 in Figure 13. 2 arcs leave from the first node to represent the next position for the batch of product contained in that section of the pipeline at time t_{next}. At any time t, there are $P+1$ arcs entering the first node of the network, that represents the first batch position in the pipeline; in order to represent whether a batch of product i enters the pipeline or the product that was in that position in the pipeline at time $t_{previous} = t - \delta$ or it keeps its position at time t. In the case of the intermediate nodes, there are 2 arcs entering and 2 arcs leaving the node. The incoming arcs represent whether the current batch of product at a given batch position inside the pipeline comes from the previous batch position or held its position from time $t_{previous}$ to time t and the leaving arcs represent whether the product will keep its position inside the pipeline or move downstream from time t to time t_{next}. The last nodes of each pipeline segment have an additional arc to represent whether the product stored in the last batch of the corresponding pipeline segment will be received or not in the last depot of the system.

Notation

Let N be the total set of indexes corresponding to the nodes modeling the different batch positions of the pipeline, $N = 1,2,\cdots,\psi$; where ψ is the total volume capacity of the pipeline given in batches.

Parameters of the Model

P : Number of products to be pumped through the pipeline

D : Number of depots

$b_i, i = 1,2,...,P$: Number of batches of product i available at the refinery

$\sum_{i=1}^{P} b_i$: Number of batches in the final sequence of products

UTW_{id}, LTW_{id} : Upper/lower time window for product i, $i = 1,2,...,P$ at depot d, $d = 1,2,...,D$

CS_d : Capacity of pipeline segment , given in batches, $d = 1,2,...,D$

$\Pi_{i,j,o,t}$: Cost of pumping 1 batch of product i from position j to position o in the pipeline at time t

$\Pi^{\rho}_{i,t}$: Cost of pumping 1 batch of product i into the pipeline from the refinery at time t

$\Pi^{\delta}_{i,d,t}$: cost of receiving operations at depots

$R_{i,d}$: Demand, in batches, of product i in depot $d, i = 1,2,\cdots P$

T : Number of discrete points in the time horizon

ψ : Total capacity of the pipeline in number of batches

TW_{id} : set of indexes t corresponding to the time window for a given product at a given depot $TW_{id} = t : LTW_{id} \leq t \leq UTW_{id}$

α_j : Set of nodes that can be visited from node j

∂_j : Set of depots that can be visited from node j

β_j : Set of nodes from which node j can be visited

ρ_{ij0}, ρ_{ijT} : Binary parameter to indicate the initial and final state of the pipeline. $\rho_{ij0} = 1$ means that at time 0 a batch of product i is occupying the j position of the pipeline.

$RHS_{i,j,l}$: Right hand side of the network conservation of flow constraints. $RHS_{i,j,0} = \rho_{i,j,0}; RHS_{i,j,T} = \rho_{i,j,T}$. $RHS_{i,j,l} = 0$ for $l \neq 0, T$.

Decision Variables

$$x_{i,l} = \begin{cases} 1 & \text{if product } i \text{ is assigned position } l \text{ in the sequence} \\ 0 & \text{otherwise} \end{cases}$$

$$y_{i,j,o,l} = \begin{cases} 1 & \text{if product } i \text{ goes from batch } k \text{ to batch } l \text{ in the pipeline from time } l \text{ to } l+1 \\ 0 & \text{otherwise} \end{cases}$$

$$z_{i,d,l} = \begin{cases} 1 & \text{if a batch of product } i \text{ enters depot } d \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$$

$$i, q = 1,2,\ldots,P;\ j = 1,2,\ldots,\sum_{i=1}^{P} b_i;\ j = 1,2,\ldots,CP;\ l = 0,1,\ldots,T$$

Objective Function

$$Inv\_Cst = \sum_{i=1}^{P} \sum_{l=1}^{T} \mathrm{I}_i^{\rho} (l-1) x_{il} \qquad \textbf{(1)}$$

$$Pump\_Cst = \sum_{i=1}^{P} \sum_{l=0}^{T-1} \sum_{j=1}^{\psi} \sum_{o \in \alpha_j} \Pi_{i,j,o,l} y_{i,j,o,l} + \sum_{i=1}^{P} \sum_{l=0}^{T-1} \Pi_{i,l}^{\rho} x_{i,l} \qquad \textbf{(2)}$$

$$Minimize Z = Pump\_Cst + Inv\_Cst \qquad \textbf{(3)}$$

Constraints

$$\sum_{l=1}^{\sum_{i=1}^{P} b_i} x_{i,l} = b_i, i = 1,2,\ldots,P \qquad \textbf{(4)}$$

$$\sum_{o \in \alpha_1} y_{i,1,o,l} + \sum_{d \in \partial_1} z_{i,d,t} - y_{i,1,1,l} - x_{i,l} = RHS_{i,1,l} \qquad \textbf{(5)}$$

$$\sum_{o\in\alpha_j} y_{i,j,o,l} + \sum_{d\in\partial_j} z_{i,d,t} - \sum_{o\in\beta_j} y_{i,o,j,l} = RHS_{i,j,l} \qquad \textbf{(6)}$$

$$\sum_{l\in TW_{id}} z_{i,d,l} = R_{i,d};\ i = 1,2,\ldots,P;\ d = 1,2,\ldots,D \qquad \textbf{(7)}$$

$$\sum_{i=1}^{P}\sum_{o\in\beta_1} y_{i,o,1,l} + \sum_{i=1}^{P} x_{il+1} = 1; l = 1,2,\ldots,T \qquad \textbf{(8)}$$

$$\sum_{i=1}^{P}\sum_{o\in\beta_j} y_{i,o,j,l} = 1;\ j = 2,\ldots,\psi; l = 1,2,\ldots,T \qquad \textbf{(9)}$$

Equations (1) and (2) account for inventory and pumping costs respectively. Equation (3) corresponds to the objective function of the model. Equation (4) enforces that only the amount available in the refinery of each product can be sent through the pipeline. Sets of constraints (5), (6) and(7) enforce the conservation of flow in the network. They include the initial, intermediate and final states of the pipeline, through the horizon planning period. Demand and time window constraints are enforced using set of constraints (7). The condition that only one product can be received in only one depot downstream the pipeline at any given point in time is enforced by set of constraints (8) and (9).

The criteria selected for the optimization usually has a direct effect on the model computational performance. In addition, some objective functions can be very difficult to implement for some event representations, requiring additional variables and complex constraints [13]. That is the case when transition costs are considered in this modeling approach. An extension providing this scenario is presented in the next chapter for which computational experiments were run and their results are reported. In the previous formulation, the proposed objective function seeks the optimization of the cost of pumping the products from the refinery through the line until they reach their destination and the transition costs.

In most real cases, a due date must be considered both for demands and productions [22]. Another situation that happens in the tank farm is the non-availability of a certain tank due to maintenance reasons. The maintenance is usually scheduled for a given day and takes place when the corresponding tank is empty [36]. Time windows should be respected in order to keep inventory management issues within operational levels. Inventory levels can increase or decrease according to the volume and flow rate of each product pumping or due to local or production consumption [27]. The entire line must be stopped if there is insufficient storage capacity at some depot to receive the specified amount of product from a batch in transit [5]. The consideration of time windows is very convenient to answer all these concerns

Model Insights

3 types of decision variables can be distinguished and so grouped in the following vectors for each product i:

$$X_i = [x_{i,1},...,x_{il},...,x_{iT}]^T \qquad \textbf{(10)}$$

$$Y_i = [\overrightarrow{y_{i,0}},...,\overrightarrow{y_{i,l}},...,\overrightarrow{y_{i,T-1}}]^T \qquad \textbf{(11)}$$

Where

$$\overrightarrow{y_{i,l}} = [y_{i,1,1,l}, y_{i,1,2,l},..., y_{i,j-1,j,l}, y_{i,j,j,l}, y_{i,j,j+1,l},..., y_{i,\psi,\psi,l}]^T \text{ for } l = 0,1,...,T-1 \qquad \textbf{(12)}$$

and

$$Z_i = [\overrightarrow{z_{i,1}},...,\overrightarrow{z_{i,l}},...,\overrightarrow{z_{i,T}}]^T \qquad \textbf{(13)}$$

where

$$\overrightarrow{z_{i,l}} = [z_{i1l},...,z_{i,d,l},...,z_{i,D,l}]^T \text{ for } l = 1,...,T \qquad \textbf{(14)}$$

Vector X_i is a T -dimensional vector so the number of variables x_{il} is TP. Vector $\overrightarrow{y_{i,l}}$ is a $(2\psi-1)$ -dimensional vector and vector Y_i contains T vectors like $\overrightarrow{y_{i,l}}$. Since there is one vector like that for each product that is considered, there are $PT(2\psi-1)$ $y_{i,j,o,l}$ binary variables. Vector Z_i contains T vectors $\overrightarrow{z_{i,l}}$ each having D components. In total, there are PDT z_{idl} binary variables.

Let's define now the parameters of the objective function using matrix notation. Let $\Pi = [\vec{\Pi}_{i,0},\cdots,\vec{\Pi}_{i,l},\cdots,\vec{\Pi}_{i,T-1}]$ to be a $(2\psi-1)T$ dimensional row vector where each component

$$\vec{\Pi}_{i,l} = [\Pi_{i,1,1,l}, \Pi_{i,1,2,l},\cdots,\Pi_{i,j-1,j,l},\Pi_{i,j,j,l},\Pi_{i,j,j+1,l}\cdots,\Pi_{i,\psi,\psi,l}] \qquad (15)$$

is a $(2\psi-1)$ dimensional row vector.

Now let

$$\Pi_i^{\rho} = [\Pi_{i,1}^{\rho},\cdots,\Pi_{i,l}^{\rho},\cdots,\Pi_{i,T}^{\rho}] \qquad \textbf{(16)}$$

to be a T dimensional row vector, which components correspond to the cost of pumping one batch of product i from the refinery in position l of the sequence. The pumping costs in matrix notation can be expressed as:

$$Pump\_Cst = \sum_{i=1}^{P}\Pi_i Y_i + \sum_{i=1}^{P}\Pi_i^{\rho} X_i \qquad \textbf{(17)}$$

For the inventory holding costs at the refinery let's consider the vector

$$\mathrm{I}_i^{\rho} = [0, \mathrm{I}_{i,2}^{\rho}, \cdots, \mathrm{I}_{i,l}^{\rho}(l-1), \cdots, \mathrm{I}_{i,T}^{\rho}(T-1)] \tag{18}$$

which is a T dimensional row vector for each $i, i = 1, \cdots, P$.

Using the previously defined parameters, the total inventory holding costs in matrix notation can be expressed as:

$$Inv\_Cst = \sum_{i=1}^{P} \mathrm{I}_i^{\rho} X_i \tag{19}$$

Now, the complete objective function in matrix notation has the following form:

$$Min Z = \sum_{i=1}^{P} \Pi_i Y_i + \sum_{i=1}^{P} \Pi_i^{\rho} X_i + \sum_{i=1}^{P} \mathrm{I}_i^{\rho} X_i + \sum_{i=1}^{P} \mathrm{I}_i Z_i \tag{20}$$

$$Min\ Z = \sum_{i=1}^{P} (\Pi_i^{\rho} + \mathrm{I}_i^{\rho}) X_i + \sum_{i=1}^{P} \Pi_i Y_i + \sum_{i=1}^{P} \mathrm{I}_i Z_i \tag{21}$$

Let $\Gamma_i = (\Pi_i^{\rho} + \mathrm{I}_i^{\rho})$, then the objective function can also be expressed as:

$$Min\ Z = \sum_{i=1}^{P} (\Pi_i^{\rho} + \mathrm{I}_i^{\rho}) X_i + \sum_{i=1}^{P} \Pi_i Y_i + \sum_{i=1}^{P} \mathrm{I}_i Z_i \tag{22}$$

Let N_i^x, N_i^y and N_i^z to be the matrices of coefficients for the vectors of variables X_i, Y_i and Z_i in the conservation of flow constraints. For the description of the general form of these matrices as well as the corresponding right hand side for the conservation of flow constraints, please see Appendix 1. Using matrix notation, the conservation of flow constraints for product i can be expressed as shown in the following equation:

$$N_i^x X_i + N_i^Y Y_i + N_i^Z Z_i = RHS_i \text{, For } i = 1,2,...,P \tag{23}$$

Let G^X and G^Y to be the matrix of coefficients of vectors X_i and Y_i in sets of constraints (8) and (9). The matrix notation of these sets of constraints is given by equation (**24)** as follows:

$$G^X X_1 + G^Y Y_1 + \cdots + G^X X_i + G^Y Y_i + \cdots + G^X X_P + G^Y Y_P = 1 \tag{24}$$

There is one constraint of this type for each batch position in the pipeline at each point in time during the horizon planning period so the total number of constraints in this group is ψT.

$$
\begin{array}{llllr}
\multicolumn{5}{l}{Min\, Z = \Gamma_1 X_1 + \Pi_1 Y_1 + \mathrm{I}_1 Z_1 + \cdots + \Gamma_i X_i + \Pi_i Y_i + \mathrm{I}_i Z_i + \cdots + \Gamma_P X_P + \Pi_P Y_P + \mathrm{I}_P Z_P} \\
\multicolumn{5}{l}{Subject\, to} \\
G_1 X_1 + G_1 Y_1 & \cdots + G_i X_i + G_i Y_i & \cdots + G_P X_P + G_P Y_P & & = 1 \\
N^X X_1 + N^Y Y_1 + N^Z Z_1 & & & & = RHS_1 \\
& \ddots & & & \\
& N^X X_i + N^Y Y_i + N^Z Z_i & & & = RHS_i \\
& & \ddots & & \\
& & & N^X X_P + N^Y Y_P + N^Z Z_P & = RHS_P
\end{array}
$$

Figure 14: Structure of the problem

Appendix 2 provides insights into the general form of matrices G^X and G^Y. A block diagonal structure of the problem is presented in Figure 14 in matrix notation. The connecting constraints correspond to sets of constraints 25 and 26. The number of linking constraints as stated above is ψT. For each product there is an associated sub problem given by the network flow problem for the corresponding product defining its journey through the pipeline during the horizon planning period. The model has $P[(T+1)+D+1]+2T+T\psi+P^2(T)(\psi-1)+1$ constraints and $TP+TP(2\psi-1)+PDT+P^2T(\psi-1)$ variables.

Model Extensions

One of the most challenging features of this problem is perhaps the optimization of the transition costs. The transition costs are handled using an additional set of binary variables to model the interfaces between two consecutive batches. If a batch of product i is sequenced next to product q there is an interface between the two products i and q. The interface happens until one of the batches reaches its destination. During the period of time at which the interface occurs contamination between the two products happens as well and it generates a cost of reprocessing.

T_{iq}: Transition cost when product i is sequenced next to product q, $i, q \in \{1,2,\ldots,P\}, i \neq q$.

$$
u_{i,q,j,l} = \begin{cases} 1 & if\ product\, i\, in\, batch\, j\, is\, sequenced\, before\, product\, q\, at\, time\, l \\ 0 & otherwise \end{cases}
$$

$$
Trans\_Cst = \sum_{(i,q)} \sum_{j=1}^{\psi-1} \sum_{l=1}^{T} T_{i,q} u_{i,q,j,l} \qquad \textbf{(27)}
$$

$$
\sum_{(i,q)} \sum_{l=1}^{T} \sum_{j=1}^{\psi-1} u_{i,q,j,l} = (T)(\psi-1) - \sum_{i=1}^{P} \sum_{d=1}^{D} R_{id} \qquad \textbf{(28)}
$$

There is one additional set of variables used to model the interfaces between consecutive batches of 2 different products. Let’s consider first all the possible combinations for 2 consecutive batches of different products inside the pipeline where the first batch (the closest to the refinery) has position $j, j = 1,...,\psi - 1$ and at any point in time $l, l = 1,2,...,T$. These combinations are shown in the following matrix:

$$\mathbf{u} = \begin{bmatrix} u_{1,2} & u_{1,2} & \cdots & u_{1,q} & \cdots & u_{1,P-1} & u_{1,P} \\ u_{2,1} & u_{2,1} & \cdots & u_{2,q} & \cdots & u_{2,P-1} & u_{2,P} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ u_{i,1} & u_{i,2} & \cdots & u_{i,i} & \cdots & u_{i,P-1} & u_{i,P} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ u_{P-1,1} & u_{P-1,2} & \cdots & u_{P-1,q} & \cdots & u_{P-1,P-1} & u_{P-1,P} \\ u_{P,1} & u_{P,2} & \cdots & u_{P,q} & \cdots & u_{P,P-1} & u_{P,P} \end{bmatrix} \quad \textbf{(29)}$$

Concerning the transition costs, we can express them in matrix notation in the following manner:

$$Trans\_Cst = \sum_{(i,q):i \neq q} \mathrm{T}_{i,q} \vec{u}_{i,q} \quad \textbf{(30)}$$

The transition variables for any position inside the pipeline at any point in time can be formulated considering each entry of the previous matrix expressed for any position and any point of the horizon planning period as a vector defined as:

$$u_{i,q} = [u_{i,q,1}, \cdots, u_{i,q,l}, \cdots, u_{i,q,T}]\,\textbf{,} \quad \textbf{(31)}$$

Where

$$u_{iql} = [u_{i,q,1,l}, \cdots, u_{i,q,j,l}, \cdots, u_{i,q,\psi-1,l}], l = 1, \cdots, T \quad (32)$$

The number of components of this vector is $(\psi - 1)(T - 1)$ corresponding to $T - 1$ points in time at which the interfaces can be controlled and $(\psi - 1)$ interfaces inside the pipeline. Moreover, the total number possible combination of products is given by P^2 then, the total number of transition variables is given by $P^2 T(\psi - 1)$ and there is one constraint for each variable. Furthermore, only up to $\theta = (T)(\psi - 1) - \sum_{i=1}^{P} \sum_{d=1}^{D} R_{id}$ transition variables can be none zero variables.

All the transition variables can be expressed in a vector U with the following form:

$$U = [u_{1,1}, \cdots, u_{1,q}, \cdots, u_{1,P}, \cdots, u_{P,1}, u_{P,2}, \cdots, u_{P,q}, \cdots, u_{P,P}] \quad \textbf{(33)}$$

$$Min Z = \sum_{i=1}^{P} (\Pi_i^{\rho} + \mathrm{I}_i^{\rho}) X_i + \sum_{i=1}^{P} \Pi_i Y_i + \sum_{i=1}^{P} \mathrm{I}_i Z_i + \sum_{(i,q)} \mathrm{T}_{i,q} \vec{u}_{i,q} \quad \textbf{(34)}$$

$$Min\,Z = \Gamma_1 X_1 + \Pi_1 Y_1 + \mathrm{I}_1 Z_1 + \cdots + \Gamma_i X_i + \Pi_i Y_i + \mathrm{I}_i Z_i + \cdots + \Gamma_P X_P + \Pi_P Y_P + \mathrm{I}_P Z_P \quad + T_{11} u_{11} + \cdots + T_{1P} u_{1P} + \cdots + T_{i1} u_{i1} + \cdots + T_{iP} u_{iP} + \cdots + T_{P1} u_{P1} + \cdots + T_{PP} u_{PP}$$

Subject to

$$G_1 X_1 + G_1 Y_1 \quad \cdots + G_i X_i + G_i Y_i \quad \cdots + G_P X_P + G_P Y_P \quad = 1$$

$$N^X X_1 + N^Y Y_1 + N^Z Z_1 \quad = RHS_1$$

$$\ddots$$

$$N^X X_i + N^Y Y_i + N^Z Z_i \quad = RHS_i$$

$$\ddots$$

$$N^X X_P + N^Y Y_P + N^Z Z_P \quad = RHS_P$$

$$(T_1^1 + T_1^2) Y_1 + T_Z Z_1 \quad -2u_{11} \quad \geq 0$$

$$T_1^1 Y_1 \quad T_P^2 Y_P + T_Z Z_P \quad -2u_{1P} \quad \geq 0$$

$$T_1^2 Y_1 + T_Z Z_1 \quad + T_i^1 Y_i \quad -2u_{i1} \quad \geq 0$$

$$T_i^1 Y_i + \quad T_P^2 Y_P + T_Z Z_P \quad -2u_{iP} \quad \geq 0$$

$$T_1^2 Y_1 + T_Z Z_1 \quad T_P^1 Y_P \quad -2u_{P1} \quad \geq 0$$

$$(T_P^1 + T_P^2) Y_P + T_Z Z_P \quad -2u_{PP} \quad \geq 0$$

$$u_{11} + \cdots + u_{1P} + \cdots + u_{i1} + \cdots + u_{iP} + \cdots + u_{P1} + \cdots + u_{PP} = \theta$$

Figure 15: Problem structure considering interfaces

CHAPTER V

SOLUTION PROCEDURE AND COMPUTATIONAL EXPERIMENTS

Introduction

In this chapter, the solution procedures that were considered for this problem are presented: a decomposition based solution scheme and a branch and bound procedure, were several schemes for assigning priorities for branching are presented, and explored in the following chapter via the computational experiments. The decomposition scheme for integer problems is known in the literature as *branch and price*. The Branch and Price approach have been a successful procedure to solve integer programs with special structures. However, the computational results obtained for the problem under consideration in this dissertation discourage its implementation. In spite of the network structure of the sub problems (column generation problems), perhaps the size of the master problem precludes the use of this decomposition approach in the solution procedure for this model. On the other hand, the branch and bound algorithm outperformed the decomposition approach by far. Branching priorities are determined based on the dynamic aspects of the problem for the branch and bound algorithm and their interpretation are commented. In all the sets of problems, running the MIP CPLEX solver with the default settings was outperformed by one or more of the schemes to assign priorities for branching.

A Decomposition Based Solution Procedure.

Decomposition approaches have been widely applied to solve large scale optimization problems with special structures. The structure of the mathematical model introduced in the last chapter suggests the implementation of a Branch and Price approach, which couples a Dantzig-Wolfe decomposition algorithm and a branch and bound algorithm by applying the decomposition principle at each node of the branch and bound tree. Branch and Price is the generalization of a branch and bound algorithm that includes the generation of columns solving the pricing problem. The column generation is applied throughout the branch and bound tree prior to branching. Branching occurs when no profitable columns can be found and the LP solution does not satisfy the integrality conditions. This approach has been implemented for large scale integer programs with binary variables when special structures are present. Perhaps the main reason to apply a branch and price strategy is the quality of the bound in the branch and bound tree that can be obtained using this approach, more than the computational time required to solve the problem as reported in the literature.

Problem Decomposition.

A decomposition strategy divides an intractable problem into smaller, less challenging sub problems, develops solutions for the sub problems and assembles them into the master problem to generate an optimal solution for the original problem [43]. Given the network structure of the system under consideration, each sub problem is a minimum cost network flow problem, which can be solved very efficiently using the network simplex method. The problem structure, in matrix notation, is as presented in Figure 14. Let $\Delta_i = [\Gamma_i, \Pi_i, \mathrm{I}_i]$, $\Phi_i = [X_i, Y_i, Z_i]^t$, $G_i = [G_i^x, G_i^y, 0]$, $N = [N^x, N^y, N^z]$, $\mathrm{B} = [RHS_1, \cdots, RHS_i, \cdots, RHS_P]^t$. Under these new parameter definitions, the problem can then be expressed as follows:

$$
\begin{aligned}
& MinZ = \sum_{i=1}^{P} \Delta_i \Phi_i \\
& S.t. \\
& \sum_{i=1}^{P} G_i \Phi_i = 1 \\
& N\Phi_i = \mathrm{B}_i, i = 1,2,\cdots,P
\end{aligned}
$$

Let's consider the polyhedral corresponding to the network sub problem for product i $N\Phi_i = \mathrm{B}_i, i = 1,2,\cdots,P$. Any point Φ can be expressed as a convex combination of the extreme points of the polyhedron.

$$\Phi = \sum_{k=1}^{t_i} \lambda_{ik} \Phi_i$$

$$\sum_{k=1}^{t_i} \lambda_{ik} = 1,$$

Where t_i is the number of extreme points in the polyhedron of the network sub problem. The problem can then be reformulated as:

$$MinZ = \sum_{i=1}^{P} \sum_{k=1}^{t_i} \Delta_i \Phi_i \lambda_{ik}$$

Subject to

$$\sum_{i=1}^{P} \sum_{k=1}^{t_i} G_i \Phi_i \lambda_{ik} = 1$$

$$\sum_{k=1}^{t_i} \lambda_{ik} = 1 \text{ for } i = 1,2,\cdots,P$$

$$\lambda_{ik} \geq 0, \quad i = 1,2,\cdots,P\ , \ k = 1,2,\cdots,t_i$$

The linking constraints correspond to the set of constraints 35 and 36. There are $\Psi\mathrm{T}$ constraints of this type, where Ψ is the number of batches in which the pipeline is divided and T is the number of time periods in the planning horizon. There are P network sub problems, one for each product. In order to solve the restricted master problem, two phases are considered in the procedure: phase I, minimizes the sum of the artificial variables in the master problem, added to obtain an initial feasible solution for the linear relaxation of the problem. In phase II, the objective function of the original problem is optimized. New columns are generated at each iteration from the sub problems and added to the pool of columns in the restricted master problem to be solved again. Let $(\mu, \beta_1, \cdots, \beta_P)$ to be the vector of dual variables of the linking constraints and the convexity constraints in the restricted master problem. Let φ to be a vector of artificial variables added to the linking constraints of the master problem which sum has to be minimized during the phase I of the algorithm. The master problem for the phase I is presented below:

$$MinZ = \sum_{j=1}^{\psi}\sum_{t=1}^{T}\varphi_{jt}$$

Subject to

$$\sum_{i=1}^{P}\sum_{k=1}^{t_i} G_i \Phi_i \lambda_{ik} + \varphi = 1$$

$$\sum_{k=1}^{t_i}\lambda_{ik} = 1 \ \text{ for } \ i = 1,2,\cdots,P$$

$$\lambda_{ik} \geq 0, \quad i = 1,2,\cdots,P\ , \ k = 1,2,\cdots,t_i$$

$$\varphi_{jt} \geq 0, j = 1,2,\cdots,\gamma; t = 1,2,\cdots,T$$

The network flow sub problem for product i during phase I is:

$$Max\ Zsub = \mu G_i \Phi_i + \beta_i$$

Subject to

$$N\Phi_i = \mathrm{B}_i$$

The network flow sub problem for product i during phase II is:

$$Max\ Zsub = (\mu G_i - \Delta_i)\Phi_i + \beta_i$$

Subject to

$$N\Phi_i = \mathrm{B}_i$$

The branch and price solution procedure first generates the root node of the branch and bound tree by articulating the solution of the previous sub problems in a Dantzig-Wolfe Decomposition algorithm.

Dantzig-Wolfe Algorithm: *Generates the root node in the branch and bound tree of the Branch and Price Algorithm*

Start

$phase = 1$

Step 1: $\mu = 0$, $\beta_i = 0$, *solve the network sub problems for phase II-initial proposals*

Step 2: solve the restricted master problem for phase I

While $Zmaster > 0$

Step 3: obtain duals and solve the network sub problems for phase I

Step 4: If $SubZ_i > 0$ *then accept proposal from product* i

Step 5: If no new proposals and $Zmaster$ *>0 then abort: original problem is infeasible*

Step 6: solve the new restricted master problem for phase I

End While

$phase = 2$

Step 7: solve the restricted master problem for phase II

While New Proposals Available

Step 6: obtain duals and solve the network sub problems for phase II

Step 7: If $SubZ_i > 0$ *then accept new proposal from product* i

Step 8: If no new proposals then

Current solution for master problem is optimal

Else

Solve the new restricted master problem for phase II

End if

End While

End

This algorithm creates the root node of the branch and bound tree in the branch and price algorithm. Once this node is created and no integral solution is at hand, the branching process starts. Branching can happen in 2 different ways: on original variables or in the new variables (the master problem variables).

The pricing problem has to be adequate with the branching strategy. For instance, if one chooses a given variable $\lambda_{i,k}$, the two possible branches in the branch and bound tree are $\lambda_{i,k} \leq 0$ and $\lambda_{i,k} \geq 1$. In the first case, the pricing sub problem for product i has to restrict the variables $x_{i,t}, y_{i,j,k,t}, z_{i,d,t}$ that are equal to one in the k^{th} column to be equal to zero and in the second case; the corresponding variables are restricted to be ≥ 1 for product i. This additional set of constraints for the column generation sub problem destroys its network structure making it more difficult to be solved.

At each of the new nodes, columns are generated and added to the master problem until no new columns improving the master objective function can be generated. In a branch and price algorithm, the Dantzig-Wolfe decomposition algorithm is applied at each node of the branch and bound tree. Therefore, the implementation of such an algorithm requires the implementation of the Dantzig-Wolfe decomposition approach to solve the problem coupled with a branch and bound algorithm.

A Branch and Bound Approach

In previous sections, the complicating constraints of the mathematical model were justified with the need to enforce the dynamic aspects of the pipeline, e.g.: If batch B is injected after batch A in

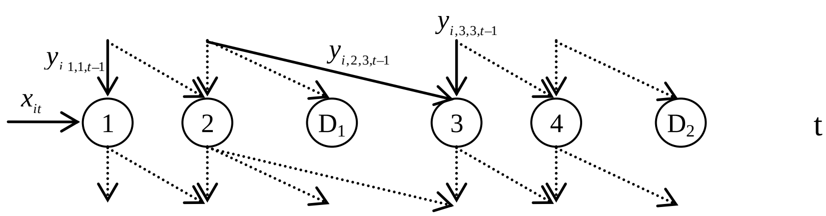

Figure 16: Only one batch of product can occupy a batch position inside the line.

the refinery, this set of constraints guarantees that at a later point in time that sequence will be preserved. The logic says that batch B cannot show up downstream batch A at some point in time if it was injected after batch A. In order to better illustrate this set of constraints, consider nodes 1 and 3 in Figure 16 to provide two examples of this set of constraints. In both cases, only one bold arc can enter the node. This is modeled for the two nodes respectively as follows:

$$\sum_{i=1}^{P} x_{it} + \sum_{i=1}^{P} y_{i,1,1,t-1} = 1 \qquad \textbf{(37)}$$

$$\sum_{i=1}^{P} y_{i,2,3,t-1} + \sum_{i=1}^{P} y_{i,3,3,t-1} = 1 \qquad \textbf{(38)}$$

Equations 37 and 38 enforce, in both cases, that only one batch of product can occupy each batch position inside the pipeline. Although perhaps at first sight it is not evident, they also imply that if the optimal integral values of x and z are available, then the optimal values for y will be integral without enforcing this condition. For a big scenario of the problem, the size of this set of constraints precludes the use of a decomposition approach despite the network structure of the column generation problem. The master problem becomes a huge problem with prohibitive solving computational times when large scenarios of the problem are considered. Nonetheless, this set of constraints can be exploited in a branch and bound implementation by relaxing the integrality of the set of variables y, making the use of computational time during the solution procedure more efficient. It turns out that this is the biggest set of binary variables in the problem. In order to illustrate this, let´s recall some important features of the proposed mathematical model: 3 types of binary variables can be distinguished in the network: x, y and z. The x variables are used to decide the sequence and the amounts in which the different products are going to be injected into the pipeline. The z variables are employed to decide whether or not a given batch of product is received at a given depot during the horizon planning period. The y variables on the other hand, are used to describe the journey of the different batches of products through the pipeline during the horizon planning period. Those variables are dependent on the values taken by the other two types of variables due to the set of complicating constraints. Once the optimal integral values of the x and z variables are at hand, the integral optimal values of the y variables are available as well, thanks to this set of constraints. The point now is how to determine the values of x and z using a branch and bound approach in a way that the use of computational time is as efficient as possible.

Branching Schemes.

Based on the logic of the operation of the system, several branching schemes for these two sets of variables were explored, however only the 4 most efficient in the use of the computational time are reported in this dissertation. The logic in the operation of the system is employed in the decision to assign priorities for branching in the x and z variables. This logic can be viewed in different ways, for instance: in a normal system operation it may be first decided what product, -and in which amount-, is going to be injected into the line and then to decide what is going to happen with the products that are inside the pipeline, and this means to decide what is the depot located downstream the line that is going to receive product and at what point in time this is going to happen. Under this scheme of operation, the x variables have bigger priorities for branching than the z variables. This feature limits the number of possible branching variables to as many as the number of depots along the line times the number of points in the horizon planning period plus the number of time periods. A brute force type of branch and bound solution procedure would consider a significantly bigger number of variables as candidates to perform branching and would not decide the branching priorities in an informed fashion considering the dynamic aspects of the system, not to mention the consideration of the integrality conditions for the y variables.

The computational experiments for 4 branching schemes are reported. Other branching schemes were also tested but their poor performance hindered them from being reported. The computational results are compared with the default settings for the CPLEX solver modeled via GAMS and without relaxing the binary nature of the y variables. The branching variables as well as priorities to pick them and their justification are the key aspects that define this branch and bound scheme.

Branching Scheme 1

For $t = 1,2,\cdots,T$, the set of variables x_{it} is given the highest priorities and then the set of variables z_{idt} are assigned decreasing priorities in an order determined chronologically and based on the location of depots downstream the line. For a better illustration, ordering the variables based on their branching priorities from highest to lowest, the variables are listed as follows:

$$x_{i1}, z_{i11}, z_{i21}, \cdots, z_{iD1}, x_{i2}, z_{i12}, z_{i22}, \cdots, z_{iD2}, \cdots, x_{iT}, z_{i1T}, z_{i2T}, \cdots, z_{iDT}$$

Under this approach, from the first point in the horizon planning period to the last one, it is first determined whether or not a given product is injected into the line at a given point in time and then, the procedure identifies the depot that receives a batch of product downstream the line. The product acceptance is evaluated for those z variables with non-integer values in the order in which the depots are located through the line, e.g.: first, branch on z_{i11}, then branch on z_{i21}, and so on, if their values are not integral. A pseudo code determining the branching priorities of the variables under this branching scheme is presented below:

```
For t = 1,2,...,T
    Branch on x_it
    For d = 1,2,...,D
        Branch on z_idt
    Next d
Next t
```

Branching Scheme 2

This scheme selects as branching variables the set x_{it} in chronological order from the first to the last point in the horizon planning period. Once the integral values for the x_{it} variables are determined, branching on the z_{idt} variables starts in chronological and depot location order, given higher priorities at earlier points in time and depots with closer locations to the refinery. Ordering the variables based on their branching priorities from highest to lowest, the variables are listed as follows:

$$x_{i1}, x_{i2}, \cdots, x_{iT}, z_{i11}, z_{i21}, \cdots, z_{iD1}, z_{i12}, z_{i22}, \cdots, z_{iD2}, \cdots, z_{i1T}, z_{i2T}, \cdots, z_{iDT}$$

This scheme first determines the sequence in which the different batches of products have to be pumped and then it identifies, in chronological order, the depot that accepts product downstream the line, evaluating first those that are closer to the refinery.

A pseudo code of this branching scheme is presented below:

For $t = 1,2,\cdots,T$
    Branch on x_{it}
Next t
For $d = 1,2,\cdots,D$
    Branch on z_{idt}
Next d

Branching Scheme 3

This scheme operates in the opposite way to branching scheme 2. Branching on the z_{idt} variables occurs first on a chronological and depot location basis, assigning higher priorities to the z_{idt} variables corresponding with earlier points in time and depots with closer locations to the refinery and considering those with later points in time and away locations from the refinery in the last stages of the optimization process. Once the integral values for the z_{idt} variables are determined, branching on the x_{it} variables starts assigning priorities for branching from highest to lowest in chronological order. Ordering the variables based on their branching priorities from highest to lowest, the variables are listed as follows:

$$z_{i11}, z_{i21}, \cdots, z_{iD1}, z_{i12}, z_{i22}, \cdots, z_{iD2}, \cdots, z_{i1T}, z_{i2T}, \cdots, z_{iDT}, x_{i1}, x_{i2}, \cdots, x_{iT}$$

Under this scheme, the algorithm first identifies the depot that accepts a batch of product at each point in the horizon planning period in chronological order and then, based on that, the sequence of products to be pumped into the line is determined. A pseudo code of this branching scheme is presented below:

For $d = 1,2,\cdots,D$
    Branch on z_{idt}
Next d
For $t = 1,2,\cdots,T$
    Branch on x_{it}
Next t

Branching Scheme 4

This scheme operates in an opposite way in which the first branching scheme does. In chronological order for each discrete point in the horizon planning period, it first determines the integral values of the z_{idt} variables and then it establishes the values of the x_{it} variables. The integral values for z_{idt} are determined in the same order in which the depots are located downstream the line in chronological order. Once the process finishes determining the z_{idt}, the values for the x_{it} variables are then determined in a chronological order.

Ordering the variables based on their branching priorities from highest to lowest under this branching scheme, the variables are listed as follows:

$$z_{i11}, z_{i21}, \cdots, z_{iD1}, x_{i1}, z_{i12}, z_{i22}, \cdots, z_{iD2}, x_{i2}, \cdots, z_{i1T}, z_{i2T}, \cdots, z_{iDT}, x_{iT},$$

Under this scheme, in chronological order, the depot that accepts product at a given point in time is first identified and then the product that is injected into the line at that point in time is determined.

The corresponding pseudo code is given as follows:

For $t = 1, 2, \cdots, T$
    For $d = 1, 2, \cdots, D$
        Branch on z_{idt}
    Next d
    Branch on x_{it}
Next t

Other branching schemes that assign the higher branching priorities to the variables corresponding with the last points in time and the lowest branching priorities to the variables at the early stages of the horizon planning were also explored. It was found that these schemes had poorer computational performances.

CHAPTER VI
COMPUTATIONAL EXPERIMENTS AND ANALYSIS OF RESULTS

Introduction

In this chapter, the results of the computational experiments for a decomposition solution scheme for this problem are presented in order to expose its poor performance, in spite of the network structure of the column generation sub problems, compared with the branch and bound approach. The goal of the computational experiments is to compare the performance of the branching schemes that are being considered in terms of computational time, solution quality and robustness for the problem of interest.

Computational Experiments for the Decomposition Scheme

A decomposition scheme is presented and the results of the computational experiments that were run show that the time to create the root node of the tree, even for small instances of the problem, is outperformed by the MIP CPLEX Solver modeled in GAMS and becomes prohibitive when mid size scenarios of the problem are considered. This discourages the implementation of a branch and price procedure since this computational time is a lower bound of the computational time required to solve the problem using the Branch and Price algorithm.

		LP Relaxation		MIP Default Settings			DW LP Relaxation		
Instance	Topology	Objective	Time	Objective	Time	Sol/Gap	Objective	Time	Solution
1	1/2/2/s/u/6	2267,172	0,156	2267,172	0,281	Optimal	2267,172	29,141	Optimal
2	1/2/2/s/u/8	4601,174	0,188	4601,174	0,344	Optimal	4601,174	91,093	Optimal
3	1/2/2/s/u/10	5512,269	0,187	5512,269	0,219	Optimal	5512,269	116,797	Optimal
4	1/2/2/s/u/14	11683,69	0,219	11683,69	0,235	Optimal	11683,69	424,891	Optimal
5	1/2/2/s/u/16	13592,34	0,219	13627,932	0,25	Optimal	13592,34	228,812	Optimal
6	1/2/2/s/u/18	16870,581	0,219	16876,573	0,235	Optimal	16870,581	499,704	Optimal
7	1/2/2/s/u/20	18698,789	0,266	18701,212	0,671	Optimal	18698,789	1161,265	Optimal
8	1/2/2/s/u/24	30387,932	0,296	30505,555	0,297	Optimal	30387,932	670,985	Optimal
9	1/2/2/s/u/28	36668,044	0,266	38380,761	0,297	Optimal	36668,044	1473,563	Optimal
10	1/2/2/s/u/34	50969,962	0,328	50972,421	0,282	Optimal	50969,962	5284,485	Optimal
11	1/2/2/s/u/40	36668,044	0,344	38380,761	0,532	Optimal	36668,044	1691,406	Optimal

Table 7: Computational Time for the LP Relaxation, MIP solver and DW LP Relaxation

The maximum computational time allowed for the solver to run in each instance is 5 hours, which is the minimum amount of time reported in the literature as available to make a decision for this problem. The results are displayed in Table 7. The instances correspond to different scenarios of a system with 1 refinery pumping 2 products through a single unidirectional pipeline towards 2 depots. The variation is in the length of the horizon planning period and in all cases, the cost parameters and demand of each product at each depot is randomly determined.

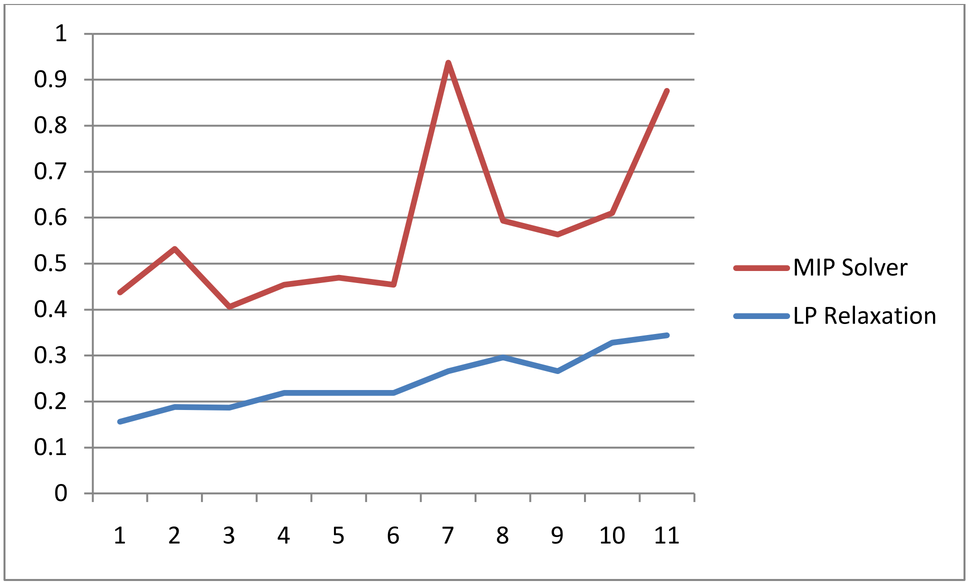

Figure 17: Computational Time in seconds for the Integer program and the LP relaxation

The computational time it takes this approach to provide the optimal solution for the linear relaxation of the problem (the root node of the branch and bound tree) is a lower bound of the computational time it takes the branch and price algorithm to provide the optimal solution for the integer problem under consideration. In order to explore the computational efficiency that can be expected from a branch and price implementation to solve this problem, computational experiments were run in a set of 11 random instances. In all cases, the computational time it took the decomposition approach to create the root node of the branch and bound tree was significantly higher than the time it took the MIP CPLEX solver to provide an optimal solution for the single binary multi commodity network flow model, and even higher than the time it took the CPLEX solver to create the root node of a branch and bound implementation.

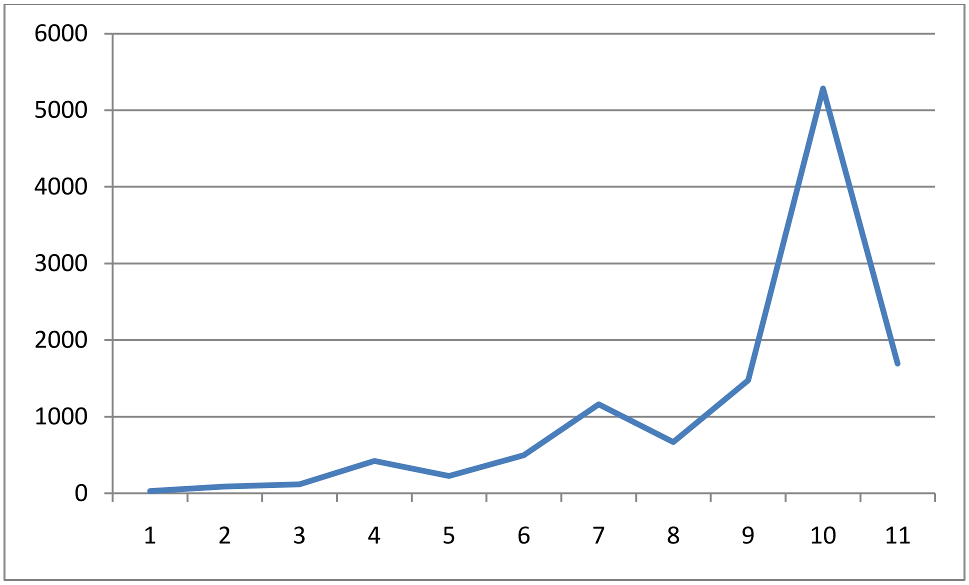

Figure 18: Computational Time in seconds for the Dantzig Wolfe Algorithm

Figure 17 plots the computational time required to solve the single model using CPLEX solver in GAMS and using the decomposition approach. For the biggest instance tested, the computational time required to solve the problem using the decomposition approach becomes prohibitive, while the solution time required using a single model is still reasonable. The time it requires for the MIP *state-of-the art* CPLEX Solver in GAMS also becomes prohibitive for large instances of the problem. The computational performance of this decomposition approach discourages the implementation of the branch and price approach which computational time is necessarily bigger than the computational time required creating the root node of the tree.

Analysis of results

The performance of a decomposition algorithm seems to be poor even for the smallest topologies that can be considered for this problem which sizes are not realistic. It is not a robust approach since the computational time can vary significantly from the expected linear trend when solving instances with a longer horizon planning period each time. It is not worthy to explore the branch and price performance under these circumstances given the poor performance of the decomposition algorithm to generate the root node of the branch and bound tree. This computational time is the lower bound of a branch and price implementation. It remains an open question what the performance is like for the branch and price approach, when the root node of the branch and bound tree is generated using a single model and then, if the solution is not integral, applying the branch and price approach. Using a pure decomposition approach is not a promising solution procedure for this model.

Computational Experiments for the Branch and Bound approach with Priorities for Branching

All branching schemes are compared with the default settings of the MIP CPLEX solver using GAMS without the relaxation of the integrality of variables y. The computational time was limited to a maximum of 5 hours and a solution within a 2% optimality gap was acceptable to stop the search process. For each branching scheme, several instances of the problem were solved with topologies ranging from 1/2/2/u/s/6 to 1/6/6/u/s/40. The parameters of the objective function as well as the different product demands were determined randomly by GAMS in order to assess robustness of the proposed procedures. Wall clock time was employed in the measure of the computational effort.

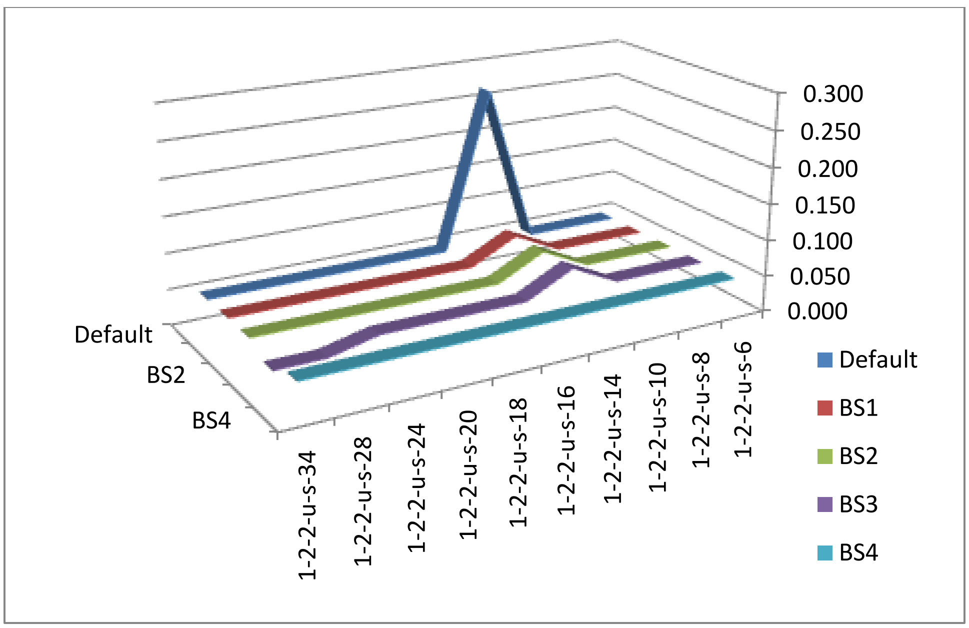

Figure 19: Computational Time Required by the Different Solution Schemes for Solving the Smallest Instances Considered.

The smallest instance of this problem consists of a single refinery, 2 depots and 2 products, as presented in previous chapters. Figure 19 provides a plot of the computational time required by the 5 different schemes to solve a family of instances where 2 depots are demanding 2 products and the horizon planning period ranges from 6 to 34 discrete points in time. Even for the smallest problem sizes, the difference in the performance among the 5 solution schemes becomes evident when, given the randomness in the selection of the parameters of the model, the number of computations necessary to solve the model is higher. The instance 1/2/2/u/s/14 requires more computational time than the others in all the solution schemes. The MIP CPLEX solver with default settings is outperformed by all the different schemes to assign priorities for branching.

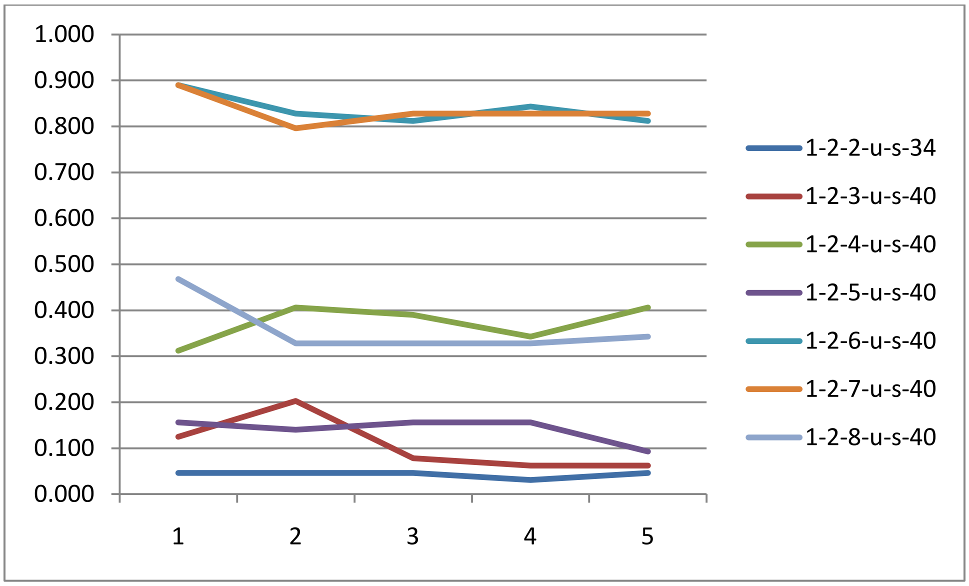

Figure 20: Computational Time Required Solving Each Instance by Every Solution Scheme

Figure 20 provides another set of instances, bigger in size than those reported in Figure 19. Again, a system with one refinery and 2 depots is considered but this time, different products, ranging from 2 to 6, have to be pumped through the line to satisfy their demand, randomly determined for each instance, at the remote depots. Furthermore, the horizon planning period has 40 discrete points in length, with the exception of the first scenario that has 34. All branching schemes outperform the default settings of the MIP CPLEX Solver, in 6 out of 7 cases. In addition to this, it also suggests that adding one product to the problem tends to increase the computational time dramatically. Presumably the longest the computational time is, the more evident the difference in the computational performance among the 5 schemes becomes. The scheme in which the MIP solver of CPLEX is run with the default settings is outperformed by the

MIP solver with the different schemes to assign priorities for branching and relaxing the integrality requirement for the y variables.

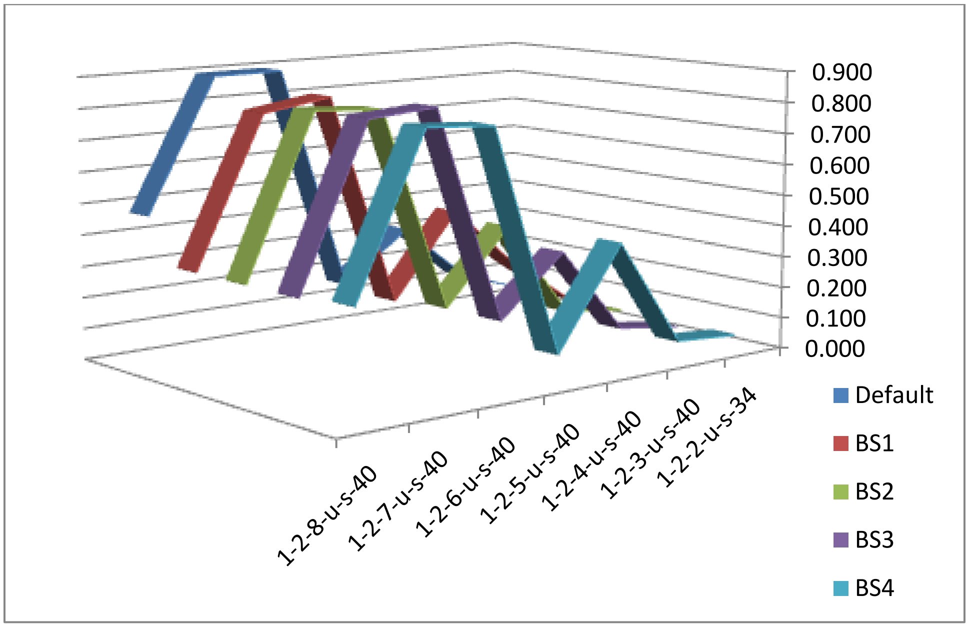

Figure 21: Computational Time Required By Each Solution Scheme to Solve Every Instance

Contrary to Figure 20, Figure 21 offers insights into the time it takes each solution scheme to solve every instance of this set. An overall look at the picture makes evident the fact that when schemes for assigning priorities for branching are considered, the MIP CPLEX Solver outperforms in all the four different cases the MIP CPLEX solver with the default settings. This is easier to observe for bigger sizes of the model. The differences become more visible when the number of products is increased.

Table 8: Results of the Computational Experiments for the GAMS Default Settings

	Instance	Equations	Variables	Discrete Variables	Iterations	Nodes	Objective Function	Relative Optimality Gap %	Absolute Optimality Gap	Gams Default Settings
1	1-2-2-u-s-6	87	121	120	32	0	2,525.453000	0	0	0.046
2	1-2-2-u-s-8	111	161	160	64	0	4,057.207000	0	0	0.046
3	1-2-2-u-s-10	135	201	200	85	0	5,193.911000	0	0	0.046
4	1-2-2-u-s-14	183	281	280	242	0	10,965.611000	2.634	28.888123	0.265
5	1-2-2-u-s-16	207	321	320	144	0	12,589.604000	0	0	0.046
6	1-2-2-u-s-18	231	361	360	194	0	16,554.033000	0	0	0.046
7	1-2-2-u-s-20	255	401	400	179	0	20,639.724000	0	0	0.046
8	1-2-2-u-s-24	303	481	480	232	0	28,486.152000	0	0	0.046
9	1-2-2-u-s-28	351	561	560	244	0	35,970.359000	0	0	0.046
10	1-2-2-u-s-34	423	681	680	335	0	58,591.490000	0	0	0.046
11	1-2-3-u-s-40	662	1201	1200	716	0	64,193.507000	0.075	48.173038	0.125
12	1-2-4-u-s-40	829	1601	1600	923	0	59,585.163000	0	0	0.312
13	1-2-5-u-s-40	996	2001	2000	859	0	63,550.908000	0.0748	47.532218	0.156
14	1-2-6-u-s-40	1163	2401	2400	1179	0	61,789.317000	0.3919	242.174804	0.890
15	1-2-7-u-s-50	1650	3501	3500	1550	0	87,483.437474	0.0674	58.933194	0.500
16	1-2-8-u-s-50	1857	4001	4000	4641	21	91,419.720418	0.0927	84.738066	1.265
17	1-3-2-u-s-40	1473	2161	2160	3178	10	93,176.247000	0.1548	144.209127	2.328
18	1-3-3-u-s-40	1969	3241	3240	3516	0	84,306.263000	0.8293	699.120393	1.902
19	1-3-3-u-s-50	2449	4051	4050	12517	54	130,072.123472	0.8206	1067.3903	10.810
20	1-3-4-u-s-50	3065	5401	5400	99980	330	124011.6477	1.9724	2446.00656	77.765
21	1-3-5-u-s-50	3681	6751	6750	4223683	19249	124,496.810013	1.9777	2462.1142	2664.703
22	1-3-6-u-s-50	4297	8101	8100	1951229	5202	122,670.254677	1.9851	22435.1413	1588.313
23	1-4-2-u-s-50	2443	3601	3600	5282	0	139,927.370028	1.0394	1454.38748	5.015
24	1-4-3-u-s-80	5184	8641	8640	1093616	2765	327420.2793	1.9814	6487.51589	1205.062
25	1-4-5-u-s-50	4906	9001	9000	3712068	8553	138175.6579	1.9756	2729.78695	4818.438
26	1-4-6-u-s-50	5727	10801	10800	14538973	23620	130981.9656	4082.29274	3.1167	18000.000
27	1-5-2-u-s-50	3053	4501	4500	367641	2464	154612.5488	2429.81655	1.5716	276.562
28	1-5-3-u-s-50	4079	6751	6750	16175439	66204	160303.6831	3188.73932	1.9892	12763.078
29	1-3-4-u-s-40	2465	4321	4320	NA	NA	Inf/Unb	N/A	N/A	171.297
30	1-4-3-u-s-40	2624	4321	4320	NA	NA	Inf/Unb	NA	NA	549.156
31	1-4-3-u-s-50	4079	6751	6750	NA	NA	Inf/Unb	NA	NA	18000
32	1-4-4-u-s-40	3285	5761	5760	NA	NA	Inf/Unb	N/A	N/A	405.454

Table 9: Results of the Computational Experiments for the Scheme 1 to Assign Priorities for Branching

	Instance	Equations	Variables	Discrete Variables	Iterations	Nodes	Objective Function	Relative Optimality Gap %	Absolute Optimality Gap	Branching Scheme 1
1	1-2-2-u-s-6	87	121	36	35	0	2,525.453184	0.0505	1.275733	0.109
2	1-2-2-u-s-8	111	161	48	71	0	4,057.206569	0	0	0.125
3	1-2-2-u-s-10	135	201	60	82	0	5,193.911350	0	0	0.141
4	1-2-2-u-s-14	183	281	84	228	6	10,976.919547	62.477534	0.5692	0.156
5	1-2-2-u-s-16	207	321	96	158	0	12,589.604217	0	0	0.125
6	1-2-2-u-s-18	231	361	108	193	0	16,554.033006	0	0	0.125
7	1-2-2-u-s-20	255	401	120	165	0	20,639.733513	0	0	0.140
8	1-2-2-u-s-24	303	481	144	255	0	28,486.151580	0	0	0.140
9	1-2-2-u-s-28	351	561	168	236	0	35,970.358839	0	0	0.141
10	1-2-2-u-s-34	423	681	204	309	0	58,591.489680	0	0	0.203
11	1-2-3-u-s-40	662	1201	360	597	0	64,201.097729	0.1266	81.303709	0.172
12	1-2-4-u-s-40	829	1601	480	1537	14	59,670.997416	0.1746	104.212622	0.594
13	1-2-5-u-s-40	996	2001	600	812	0	63,507.847572	0.007	4.472205	0.234
14	1-2-6-u-s-40	1163	2401	720	1187	0	61,958.812859	0.6744	417.868095	0.968
15	1-2-7-u-s-50	1650	3501	1050	1550	0	87483.43747	0.0674	58.933194	0.359
16	1-2-8-u-s-50	1857	4001	1200	4005	20	91563.80938	0.2502	229.058648	2.156
17	1-3-2-u-s-40	1473	2161	320	6113	75	93,726.606279	0.873	818.261377	2.375
18	1-3-3-u-s-40	1969	3241	480	17128	120	84,310.010582	0.9504	801.242253	13.500
19	1-3-3-u-s-50	2449	4051	600	258257	2153	131,066.620600	1.5821	2,074	146.343
20	1-3-4-u-s-50	3065	5401	800	3997672	11438	123018.0133	1.1301	1390.27339	2411.437
21	1-3-5-u-s-50	3681	6751	1000	14984194	54092	124,324.002935	1.8872	2346.1806	16303.015
22	1-3-6-u-s-50	4297	8101	1200	9715351	22184	125,656.431556	5553.83237	4.4199	18000.000
23	1-4-2-u-s-50	2443	3601	500	3335	0	140,122.811390	1.3485	1889.5041	9.094
24	1-4-3-u-s-80	5184	8641	1200	5258175	10780	369172.1659	13.1065	48385.6282	18000.000
25	1-4-5-u-s-50	4906	9001	1250	763166	1075	137519.0627	1.7917	2463.98197	1043.359
26	1-4-6-u-s-50	5727	10801	1500	11792236	13488	156433.5247	19.2484	30110.875	18000.000
27	1-5-2-u-s-50	3053	4501	600	123404	757	153813.234	1.3225	2034.23468	250.218
28	1-5-3-u-s-50	4079	6751	900	12257822	50657	159372.3043	1.602	2553.1515	10204.234
29	1-3-4-u-s-40	2465	4321	640	NA	NA	Inf/Unb	NA	NA	18000.000
30	1-4-3-u-s-40	2624	4321	4320	NA	NA	Inf/Unb	NA	NA	18000.000
31	1-4-3-u-s-50	4079	6751	750	NA	NA	Inf/Unb	NA	NA	18000.000
32	1-4-4-u-s-40	3285	5761	800	NA	NA	Inf/Unb	NA	NA	18000.000

Table 10: Results of the Computational Experiments for the Scheme 2 to Assign Priorities for Branching

	Instance	Equations	Variables	Discrete Variables	Iterations	Nodes	Objective Function	Relative Optimality Gap %	Absolute Optimality Gap	Branching Scheme 2
1	1-2-2-u-s-6	87	121	36	35	0	2525.453184	0.0505	1.275733	0.109
2	1-2-2-u-s-8	111	161	48	71	0	4057.206569	0	0	0.093
3	1-2-2-u-s-10	135	201	60	82	0	5193.91135	0	0	0.109
4	1-2-2-u-s-14	183	281	84	243	7	10975.8731	0.4776	52.419879	0.062
5	1-2-2-u-s-16	207	321	96	158	0	12589.60422	0	0	0.109
6	1-2-2-u-s-18	231	361	108	193	0	16554.03301	0	0	0.109
7	1-2-2-u-s-20	255	401	120	165	0	20639.72351	0	0	0.093
8	1-2-2-u-s-24	303	481	144	255	0	28486.15158	0	0	0.093
9	1-2-2-u-s-28	351	561	168	236	0	35970.35884	0	0	0.093
10	1-2-2-u-s-34	423	681	204	309	0	58591.4897	0	0	0.093
11	1-2-3-u-s-40	662	1201	360	597	0	64201.09773	0.1266	81.303709	0.140
12	1-2-4-u-s-40	829	1601	480	1330	10	59731.17373	0.2752	164.388938	0.640
13	1-2-5-u-s-40	996	2001	600	812	0	63507.84757	0.007	4.472205	0.171
14	1-2-6-u-s-40	1163	2401	720	1187	0	61958.81286	0.6744	417.868095	1.422
15	1-2-7-u-s-50	1650	3501	1050	1550	0	87483.43747	0.0674	58.933194	0.359
16	1-2-8-u-s-50	1857	4001	1200	4406	20	91606.11459	0.2962	271.363856	2.343
17	1-3-2-u-s-40	1473	2161	320	2529	10	93605.10452	0.7444	696.759618	2.422
18	1-3-3-u-s-40	1969	3241	480	9211	50	85110.16919	1.8875	1606.45887	14.718
19	1-3-3-u-s-50	2449	4051	600	11942	49	129901.0121	0.765	994	22.171
20	1-3-4-u-s-50	3065	5401	800	109578	320	123,294.129788	1.4406	1776.19571	113.406
21	1-3-5-u-s-50	3681	6751	1000	1820501	7412	124,117.730396	1.7954	2228.35501	4560.093
22	1-3-6-u-s-50	4297	8101	1200	400843	1159	122,177.468284	1.8114	2213.097	1802.000
23	1-4-2-u-s-50	2443	3601	500	3335	0	140,122.811390	1.3485	1889.5041	17.546
24	1-4-3-u-s-80	5184	8641	1200	64578	130	325321.0097	1.4529	4726.46025	225.937
25	1-4-5-u-s-50	4906	9001	1250	763166	1075	137519.0627	1.7917	2463.98197	2196.734
26	1-4-6-u-s-50	5727	10801	1500	7446133	10039	129426.0168	2.3918	3095.64128	18000
27	1-5-2-u-s-50	3053	4501	600	123404	757	153813.234	1.3225	2034.23864	207.843
28	1-5-3-u-s-50	4079	6751	900	47779	99	159735.2516	1.9357	3091.98752	225.343
29	1-3-4-u-s-40	2465	4321	640	NA	NA	NA	NA	NA	18000.000
30	1-4-3-u-s-40	2624	4321	4320	NA	NA	Inf/Unb	NA	NA	18000.000
31	1-4-3-u-s-50	4079	6751	750	NA	NA	Inf/Unb	NA	NA	18000.000
32	1-4-4-u-s-40	3285	5761	800	NA	NA	NA	NA	NA	18000

Table 11: Results of the Computational Experiments for the Scheme 3 to Assign Priorities for Branching

	Instance	Equations	Variables	Discrete Variables	Iterations	Nodes	Objective Function	Relative Optimality Gap %	Absolute Optimality Gap	Branching Scheme 2
1	1-2-2-u-s-6	87	121	36	35	0	2,525.453184	0.0505	1.275733	0.109
2	1-2-2-u-s-8	111	161	48	71	0	4,057.206559	0	0	0.046
3	1-2-2-u-s-10	135	201	60	82	0	5,193.911350	0	0	0.046
4	1-2-2-u-s-14	183	281	84	243	7	10,975.873104	0.4776	52.419879	0.125
5	1-2-2-u-s-16	207	321	96	158	0	12,589.604217	0	0	0.031
6	1-2-2-u-s-18	231	361	108	193	0	16,554.033006	0	0	0.031
7	1-2-2-u-s-20	255	401	120	165	0	20,639.723513	0	0	0.031
8	1-2-2-u-s-24	303	481	144	255	0	28,486.151580	0	0	0.046
9	1-2-2-u-s-28	351	561	168	236	0	35,970.358839	0	0	0.031
10	1-2-2-u-s-34	423	681	204	309	0	58,591.489680	0	0	0.031
11	1-2-3-u-s-40	662	1201	360	597	0	64,201.097729	0.1266	81.303709	0.140
12	1-2-4-u-s-40	829	1601	480	1005	6	59,588.938154	0.029	17.302375	0.406
13	1-2-5-u-s-40	996	2001	600	812	0	63,507.847572	0.007	4.472205	0.156
14	1-2-6-u-s-40	1163	2401	720	1187	0	61,958.812859	0.6744	417.868095	0.843
15	1-2-7-u-s-50	1650	3501	1050	1550	0	87483.43747	0.0674	58.933194	0.343
16	1-2-8-u-s-50	1857	4001	1200	3969	17	91,590.689616	0.2613	239.307119	2.109
17	1-3-2-u-s-40	1473	2161	320	3651	33	94,593.952715	1.7247	1631.48262	1.578
18	1-3-3-u-s-40	1969	3241	480	16244	100	84,090.376212	0.6323	531.730616	12.781
19	1-3-3-u-s-50	2449	4051	600	9844	30	130,081.004715	0.85	1,106	14.140
20	1-3-4-u-s-50	3065	5401	800	18876	40	122,861.407016	1.0914	1340.90828	36.781
21	1-3-5-u-s-50	3681	6751	1000	233851	712	123,647.546514	1.4843	1835.29832	321.140
22	1-3-6-u-s-50	4297	8101	1200	34154	49	121,947.938579	1.6723	2039.28269	163.140
23	1-4-2-u-s-50	2443	3601	500	3335	0	140122.8114	1.3485	1889.5041	24.515
24	1-4-3-u-s-80	5184	8641	1200	59595	109	326,443.703265	1.7715	5783.10334	363.89
25	1-4-5-u-s-50	4906	9001	1250	737882	1500	137925.0845	1.9674	2713.47853	3705.984
26	1-4-6-u-s-50	5727	10801	1500	5577942	10279	130081.9712	2.2633	2944.20281	18000
27	1-5-2-u-s-50	3053	4501	600	18168	72	154445.6785	1.7298	2671.60651	84.156
28	1-5-3-u-s-50	4079	6751	900	28161	65	159208.7021	1.5867	2526.16221	134.265
29	1-3-4-u-s-40	2465	4321	640	NA	NA	NA	NA	NA	18000.000
30	1-4-3-u-s-40	2624	4321	600	NA	NA	NA	NA	NA	18000
31	1-4-3-u-s-50	4079	6751	1200	NA	NA	NA	NA	NA	18000
32	1-4-4-u-s-40	3285	5761	800	NA	NA	NA	NA	NA	18000

Table 12: Results of the Computational Experiments for the Scheme 4 to Assign Priorities for Branching

	Instance	Equations	Variables	Discrete Variables	Iterations	Nodes	Objective Function	Relative Optimality Gap %	Absolute Optimality Gap	Branching Scheme 2
1	1-2-2-u-s-6	87	121	36	35	0	2,525.453184	0.0505	1.275733	0.046
2	1-2-2-u-s-8	111	161	48	71	0	4,057.206569	0	0	0.031
3	1-2-2-u-s-10	135	201	60	82	0	5,193.911350	0	0	0.031
4	1-2-2-u-s-14	183	281	84	228	6	10,976.919547	0.5692	62.477534	0.078
5	1-2-2-u-s-16	207	321	96	158	0	12589.60422	0	0	0.015
6	1-2-2-u-s-18	231	361	108	193	0	16,554.033006	0	0	0.046
7	1-2-2-u-s-20	255	401	120	165	0	20,639.723513	0	0	0.218
8	1-2-2-u-s-24	303	481	144	255	0	28,486.151580	0	0	0.031
9	1-2-2-u-s-28	351	561	168	236	0	35,970.358839	0	0	0.046
10	1-2-2-u-s-34	423	681	204	309	0	58591.48968	0	0	0.046
11	1-2-3-u-s-40	662	1201	360	597	0	64,201.097729	0.1266	81.303709	0.140
12	1-2-4-u-s-40	829	1601	480	1537	14	59,670.997416	0.1746	104.212622	0.406
13	1-2-5-u-s-40	996	2001	600	812	0	63507.84757	0.007	4.472205	0.140
14	1-2-6-u-s-40	1163	2401	720	1187	0	61,958.812859	0.6744	417.868095	0.796
15	1-2-7-u-s-50	1650	3501	1050	1550	0	87483.43747	0.0674	58.933194	0.39
16	1-2-8-u-s-50	1857	4001	1200	4005	20	91563.80938	0.2502	229.058648	2.14
17	1-3-2-u-s-40	1473	2161	320	6375	74	93,726.606279	0.8509	797.476195	2.312
18	1-3-3-u-s-40	1969	3241	480	20975	157	85,084.663960	1.8523	1576.00061	67.843
19	1-3-3-u-s-50	2449	4051	600	455922	3869	130,102.603510	0.8332	1,084	751.765
20	1-3-4-u-s-50	3065	5401	800	3476250	10206	123,830.855630	1.7757	2198.80462	6096.156
21	1-3-5-u-s-50	3681	6751	1000	17771683	62693	124,378.891288	1.937	2409.20159	17669.890
22	1-3-6-u-s-50	4297	8101	1200	14272019	32484	132,434.804626	9.3056	12323.8892	18000.000
23	1-4-2-u-s-50	2443	3601	500	3335	0	140122.8114	1.3485	1889.5041	23.671
24	1-4-3-u-s-80	5184	8641	1200	5372883	11027	371911.9346	13.7432	51112.4202	18000
25	1-4-5-u-s-50	4906	9001	1250	8982816	10753	173581.724	22.1051	38370.4759	18000
26	1-5-2-u-s-50	3053	4501	600	1024578	10960	154863.8421	1.9565	3029.9658	2111.484
27	1-5-3-u-s-50	4079	6751	900	2458740	8148	158,655.29	1.199	1902.22558	6750.639
28	1-3-4-u-s-40	2465	4321	640	NA	NA	NA	NA	NA	18000.000
29	1-4-3-u-s-40	2624	4321	600	NA	NA	NA	NA	NA	18000
30	1-4-3-u-s-50	4079	6751	1200	NA	NA	NA	NA	NA	18000
31	1-4-4-u-s-40	3285	5761	800	NA	NA	NA	NA	NA	18000

Table 8, Table 9, Table 10 and Table 11 display the results of the computational experiments run for each solution scheme. More than 30 instances were solved in the process with different sizes and random parameters.

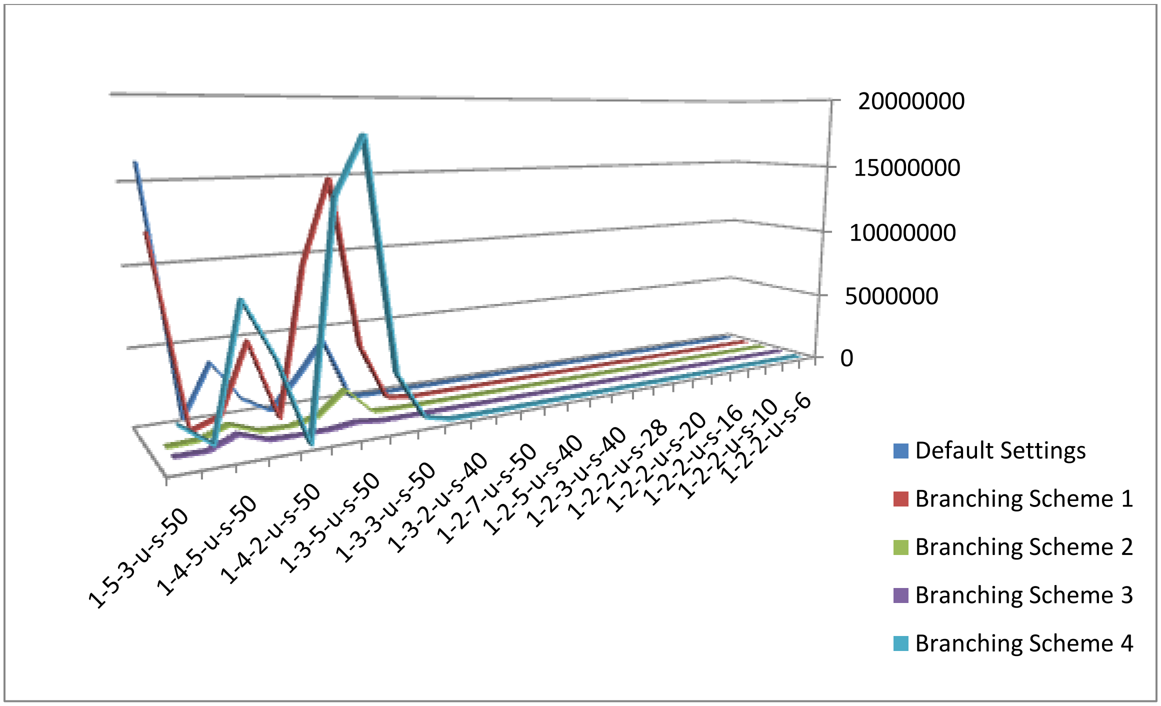

Figure 22: Number of Iterations Required by Each Solution Scheme to Solve Every Instance

Figure 22 provides a better view of the computational effort, measured in number of iterations, for each of the solution schemes under consideration to solve the instances considered in this opportunity. It is clear that, under this criterion, the scheme 3 to assign priorities for branching outperforms the other 4 solution approaches in all the instances. Perhaps, scheme 4 to assign priorities for branching is the worst solution approach for this model, being outperformed even by the default settings of the MIP CPLEX solver modeled in GAMS. The second scheme to assign priorities for branching, which is the opposite to the third one, is also the second in performance, considering the number of iterations required to find a solution.

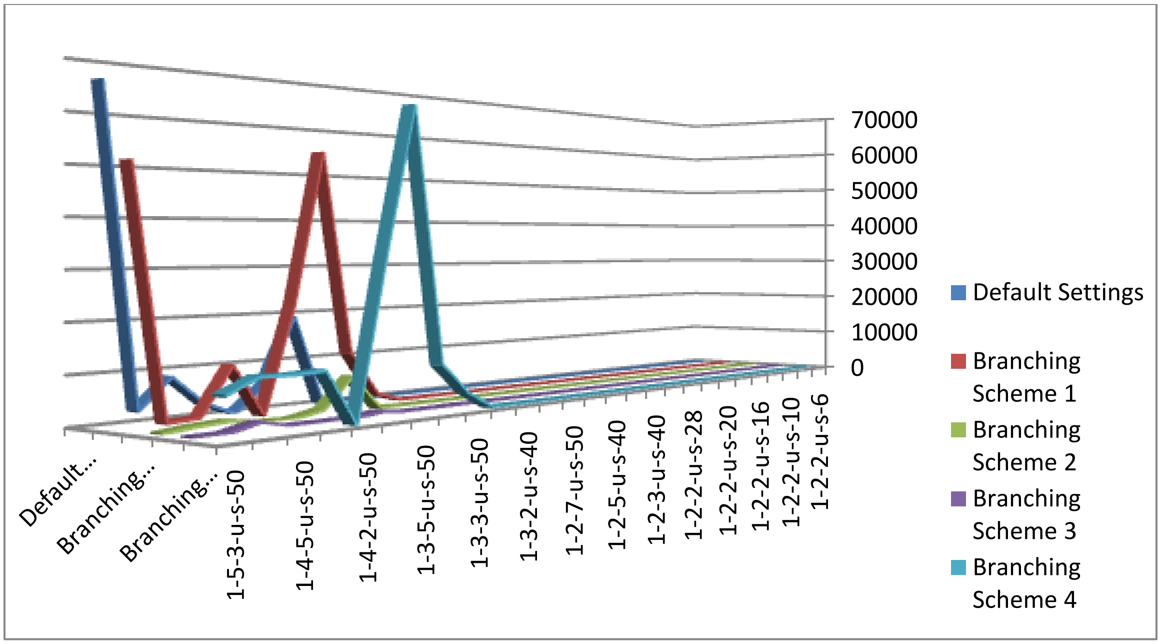

Figure 23: Number of Nodes Required by Each Solution Scheme to Solve Every Instance

Figure 23 also makes evident the superiority of scheme 3 to assign priorities for branching. This time, the number of nodes is the criterion to differentiate the performances among the different solution approaches. Again, the worst performance is for the scheme 4 to assign priorities for branching. When considering the optimality GAP, it is also evident that the scheme 3 to assign priorities for branching the non integral variables out performs all the other 4 approaches.

CHAPTER VII

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

In this chapter, a summary of this dissertation is provided, conclusions are stated and recommendations for future research directions on this field are made.

Summary

A literature review for the state of the art in pipeline transportation has been presented focusing on the transportation of refined products using pipeline systems. The most important contributions for this problem as well as their authors were highlighted. A notation for the topology of an instance for this problem was proposed creating a framework for the computational reference for the experimental stage of an interested researcher. A novel 1-0 multi commodity network flow based approach was introduced to model the system under consideration, insights into the model structure were presented and several solution schemes were explored. The best solution methodology was identified and proposed. It exploits the problem structure and the logic of the system operation in a branch and bound algorithm by reducing the number of integral variables and assigning priorities for branching. Computational experiments were run for a number of instances in order to assess the considered solution schemes in terms of solution quality, computational time and robustness. An analysis of the results obtained in the computational experiments was also provided. A solution strategy that uses information on the problem structure was found to outperform the default settings of the MIP CPLEX solver modeled with GAMS.

Conclusions

Pipeline transportation for refined products is an important problem in the distribution stage of the petroleum supply chain and more intense research efforts are required due to its importance in the petroleum industry since it transports a high volume of petroleum products using this mode of transportation.

The real life scenarios of this problem are bigger by far than the instances reported in the literature. A third part of the instances reported in the literature are random while those corresponding to real life scenarios solved within a 5% optimality GAP are single source-single pipeline systems in which 5 products are transported towards 4 depots. This problem topology is significantly smaller than real systems like Colonial’s pipeline, which serves more than 250 market zones, delivers more than 60 different products with seasonal demand and has several sources. The biggest instances reported in the literature are included in articles where the objective was not to solve the problem to optimality but to generate a feasible pumping schedule.

Perhaps the most complicating aspect in this problem might be the need to consider product contamination since in the real life problem it is an item of crucial importance due to the costs generated by contaminated product.

To model and optimize the contamination of products requires a computational time that becomes prohibitive when medium size to large instances of the problem are considered. The modeling of this product interfaces implies the use of a large set of additional variables that are defined using a new set of complicating constraints. The number of variables and the number of the corresponding constraints are proportional to the square of the number of products, the number of periods and the size of the pipeline. In real life scenarios, the size of the pipeline, the number of products and the number of batches to be sent make the mathematical model very large. The impact of a large number of products and a long pipeline can be compensated, to a small extent, by a large volume batch size.

The best scheme to solve the problem is the one that first determines, in chronological order, what is the depot that receives product at each point in time and second it determines the order of the sequence of different products that have to be sent through the pipeline.

Future Research Directions in this Field

In future research efforts, the challenge is to consider scenarios with networks of pipelines with multiple sources and bigger numbers of products to be optimized and not only to generate just a feasible solution. With the ever increasing consumption of fuels worldwide and, as it is occurring in Brazil, the appearance of more products with new specifications like bio fuels; that are also being transported via pipelines, it seems that this problem is gaining importance as its level of difficulty increases.

Product contamination was only considered in a third part of the surveyed publications and it is one of the most important issues in this type of system given the high costs of reprocessing, when it is available in site, or the cost of transporting the contaminated product back to the refinery or the nearest site were reprocessing can occur plus the costs of reprocessing. Dealing with interfaces and product contamination should be a priority in future research efforts in this field.

The topography of the terrain being traversed by the pipeline to take the products from its origin to its destination is not necessarily a flat surface. There are mountains and valleys and it makes the pressure to vary from point to point. An interdisciplinary research project to solve this problem should consider the fluid mechanics and couple it with the optimization area in order to consider all these features found in a real life scenario. Not only the location of the pumping stations is a strategic decision but their operation becomes part of this planning problem. In no previous attempts to solve this problem all this aspects have been considered.

Node selection strategies combined with a good scheme to assign priorities for branching is another solution strategy that is worthy to be investigated.

LIST OF REFERENCES

1. Alexander, V., Van, Kleve, K., 1983, “The Alaska Pipeline: a Success Story”, *Annual Review of Ecology and Systematic*, Vol. 14, pp. 443-463.
2. Breyer, Ralph, F., 1934, “Oil Pipe Lines”, *Annals of the American Academy of Political and Social Science*, Vol. 171, Banking and Transportation Problems, pp. 217-227.
3. Cafaro, D. C., Cerdá, J., 2004, Optimal scheduling of multiproduct pipeline systems using a non-discrete MILP formulation, *Computers and Chemical Engineering*, v. 28, p. 2053-2068.
4. Cafaro, D. C., Cerdá, J., 2005, Multiperiod Planning of Multiproduct Pipelines, Proceedings of the European Symposium on Computer Aided Process Engineering-15, L Puigjaner and A. Espuña Eds.
5. Cafaro, Diego C., Cerdá, Jaime, 2008, Dynamic scheduling of multiproduct pipelines with multiple delivery due dates, Computers and Chemical Engineering, 32, 728-753
6. Cafaro, Diego C., Cerdá, Jaime, 2009, Optimal Scheduling of Refined Products Pipelines with Multiple Sources, *Ind. Eng. Chem. Res.* 2009, 48, 6675-6689.
7. Camacho, E. F., Ridao, M. A., Ternero, J. A., Rodriguez, J. M., Optimal Operation of Pipeline Transportation Systems, 1991, *Automatic Control, World Congress 1990; in the service of mankind; proceedings of the 11th Triennial World Congress of the International Federation*, p. 455-461.
8. Crane, D. S., Wainwright, R. L., and Schoenefeld, D. A., 1999 Scheduling of Multiproduct Fungible Liquid Pipelines using Genetic Algorithms, *Proceedings of the 1999 ACM/SIGAPP symposium on applied computing (SAC’99), San Antonio, Texas*, p. 280-285.
9. De La Cruz, J.M., Herrán-González, A., Risco-Martín, J.L., Andrés-Toro, B., 2005, Hybrid heuristic and mathematical programming in oil pipelines networks: Use of immigrants, Journal of Zhejiang University SCIENCE, 2005 6ª(1):9-19
10. García-Sánchez, lvaro, Arreche, Luis Miguel, Ortega-Mier, Miguel; 2008, Combining Simulation and Tabu Search for Oil-derivatives Pipeline Scheduling, Studies in Computational Intelligence, (SCI)128, 301-325.
11. Emerson, Huge, N., 1950, Salient Characteristics of Petroleum Pipeline Transportation, *Land Economics*, Vol. 26, No. 1, pp. 27-39
12. Evely, R. W., 19 “The Review of Economic Studies”, Vol. 14, No. 1, pp. 16-33
13. Grossmann, Ignacio, E., and Fumarn, Kevin, C., 2009, “*Challenges in Enterprise-wide Optimization for the Process Industries*”, Optimization and Logistics Challenges in the Enterprise, Springer optimization and Its Applications, W. Chaovalitwongse et al. (eds.).

14. Emerson, H. N., 1950, Salient Characteristics of Petroleum Pipe Line Transportation, *Land Economics*, v. 26, No. 1, p. 27-39.
15. Hane C. A., Ratliff, H. D., 1995, Sequencing inputs to multi commodity pipelines, *Annals of Operations Research*, v. 57, p. 73-101.
16. Jittamai, P. Analysis of Oil Pipeline Distribution of Multiple Products Subject to Delivery Time-Windows. Ph.D. Dissertation. Department of Industrial Engineering, Texas A&M University, College Station, Texas, 2004
17. Klingman, D., Phillips, N., Steiger, D., Wirth, R., Padman, R., Krishnan R., 1987, An Optimization Based Integrated Short-Term Refined Petroleum Product Planning System, *Management Science*, v. 33, No. 7, p. 813-830.
18. Magatão, L., Arruda, L., V., R., Neves, Jr., F., 2004, A Mixed Integer Programming Approach for Scheduling Commodities in a Pipeline, Computers and Chemical Engineering 28, pp. 171-185.
19. Magatão, Leandro, Arruda, L. V. R., Neves-Jr., Flávio, 2005, Using CLP and MILP for Scheduling Commodities in a Pipeline, Proceedings of the European Symposium on Computer Aided Process Engineering, ESCAPE 15, l. Barcelona, España, May 29-June 1st. 2005
20. Maruyama Mori, Fernando, Lüders, Ricardo, Ramos de Arruda, Lúcia Valeria, Yamamoto, Lia, Bonacin, Mario Vicente, Polli, Helton Luis, Correia Aires, Mariza, Bernardo, Luis Fernando de Jesus, 2007, Simulating the Operational Scheduling of a Real-World Pipeline Network, 17th European Symposium on Computer Aided Process Engineering-ESCAPE 17.
21. Milidiú, Ruy, Pesoa, Artur Alves, Laber, Eduardo Sany, 2002, *Pipeline Transportation of Petroleum Products with No Due Dates*, In LATIN 2002: Theoretical Informatics, Springerlink-Verlag, Berlin Heidelberg 2002.
22. Milidiú, Ruy Luiz, Liporace, Federico dos Santos, de Lucena, Carlos José P., 2003, *Pipesworld: Planning pipeline transportation of petroleum derivatives*. ICAPS Workshop on Competition, Italy.
23. MirHassani, S.A., Ghorbanalizadeh, M., 2008, The multi-product pipeline scheduling system, Computers and Mathematics with Applications, 56, 891-897.
24. Moura, Arnaldo V., de Souza, Cid C., Cire, Andre A., Lopes, Tony M.T., 2008, Planning and Scheduling the Operation of a Very Large Oil Pipeline Network, Principles and Practice of

Constrained Programming, Lecture Notes in Computer Science, P.J. Stuckey (Ed.): CP 2008, LNCS 5202, pp. 36-51.

25. Moura, Arnaldo V., de Souza, Cid C., Cire, Andre A., Lopes, Tony M.T., 2008, *Heuristics and Constraint Programing Hybridizations for a Real Pipeline Planning and Scheduling Problem*, Proceedings of the 11th IEEE International Conference on Computational Science and Engineering.
26. Neves Boschetto, Suelen, Felizari, Luiz Carlos, Yamamoto, Lia, Magatao, Leandro, Stebel, Sergio Leandro, Neves-Jr., Flavio, Ramos de Arruda, Lúcia Valéria, Lüders, Ricardo, Ribas, Paulo César, Bernardo, Luis Fernando de Jesus, 2008, An Integrated Framework for Operational Scheduling of a Real-World Pipeline Network, 18th European Symposium on Computer Aided Process Engineering-ESCAPE 18.
27. Neves-Jr., Flávio, Magatão, Leandro, Stebel, Sergio Leandro, Neves Boschetto, Suelen, Felizari, Luiz Carlos, Czaikowski, Daniel Irineu, Rocha, Roger, Ribas, Paulo César, 2007, An Efficient Approach to the Operational Scheduling of a Real-World Pipeline Network, Proceedings of the 17th European Symposium on Computer Aided Process Engineering – ESCAPE17M.
28. Prewitt, R., 1942, The Operation and Regulation of Crude Oil and Gasoline Pipelines, *The Quarterly Journal of Economics*, v. 56, No. 2, p. 177-211.
29. Rejowski, Jr., R., and Pinto, J. M., 2002, An MILP Formulation for the Scheduling of Multiproduct Pipeline Systems, *Brazilian Journal of Chemical Engineering*, v. 19, No. 4, p. 467-474.
30. Rejowski, Jr., R., and Pinto, J. M., 2003, Scheduling of a Multi Product Pipeline System, *Computers and Chemical Engineering*, v. 27, p. 1229-1246.
31. Rejowski, Jr., R., and Pinto, J. M., 2004, Efficient MILP formulations and valid cuts for multiproduct pipeline scheduling, *Computers and Chemical Engineering*, v. 28, p. 1511-1528.
32. Rejowski Jr., R., Pinto, J. M., 2005, A Rigurous MINLP for the Simultaneous Scheduling and Operation of Multiproduct Pipeline Systems, Proceedings of the European Symposium on Computer Aided Process Engineering-15, L Puigjaner and A. Espuña Eds.
33. Rejowski Jr., R., and Pinto, J. M., 2008, A novel continuous time representation for the scheduling of pipeline systems with pumping yield rate constraints, *Computers and Chemical Engineering*, v. 19, No. 4, p. 467-474.

34. Relvas, Susana, Barbosa-Póvoa, Ana Paula F.D., Matos, Henrique A., Fialho, João, Pinheiro, António S., 2006, *Pipeline Scheduling and Distribution Centre Management – A Real-wold Scenario at CLC*, Proceedings of the 16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering.
35. Relvas, Susana, Matos, Henrique A., Barbosa-Póvoa, Ana Paula F.D., Fialho, Joao, Pinheiro, António S., 2006, *Pipeline Scheduling and Inventory Management of a Multiproduct Distribution Oil System*, Industrial & Engineering Chemistry Research, 45, 7841-7855.
36. Relvas, Susana, Matos, Henrique A., Barbosa-Póvoa, Ana Paula F. D., Fialho, João, 2007, *Reactive Scheduling Framework for a Multiproduct Pipeline with Inventory Management*, Industrial and Engineering Chemistry Research, 45, 5659-5672.
37. Relvas, Susana, Barbosa-Póvoa, Ana Paula F.D., Matos, Henrique A., 2009, *Heuristic batch sequencing on a multiproduct oil distribution system,* Computers and Chemical Engineering, 33, 712-730.
38. Ronen, D., 1995, Dispatching Petroleum Products, *Operations Research,* Vol. 43, No. 3, p. 379-387
39. Sasikumar, M., Prakash, P. R., Patil, S. M., and Ramani, S., 1997, PIPES: A heuristic Search Model for Pipeline Schedule Generation, *Knowledge-Based Systems*, v. 10, 169-175.
40. Shah, N., 1996, Mathematical Programming Techniques for Crude Oil Scheduling, *Computers and Chemical Engineering,* v. 20, S1227-S1232
41. Vrakas, Dimitris, Vrakas, Dimitris, Vlahavas, Ioannis Pl, 2008, Artificial Intelligence for Advanced Problem Solving Techniques, Information Science Reference, Hershey, PA.
42. Zierer, T. K., Mitchel, W. A., White, T. R., 1976, Practical Applications of Linear Programming to Shell's Distribution Problems, Interfaces, v. 6, p. 13-26.
43. Zhu, Xiaoyan, Wilhelm, Wilbert, E., 2006, Scheduling and lot sizing with sequence-dependent setup: A literature review, IIE Transactions 38, 987-1007.
44. How Pipelines Make the Oil Market Work: Their Networks, Operation and Regulation. A Memorandum Prepared for the Association of Oil Pipelines and The American Petroleum Institute's Pipeline Committee, December 2001.

APPENDIX

APPENDIX 1:

Let N_i^x to be a $(\psi(T+1)+D+1)$ by T matrix of coefficients for vector X_i in sets of constraints 4, 5, 6, and 7. The first row of the matrix corresponds to constraints 4 the following $T\psi$ rows correspond to the coefficients of variables $x_{i,l}$ in the conservation of flow constraints 5 and 6 and the remaining D rows correspond to the coefficient of the $z_{i,d,l}$ variables in set of demand constraints 7.

$$N_i^x = \begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & -1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & -1 \\ 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} \quad \textbf{(39)}$$

Let N_i^y to be $(\psi(T+1)+D+1)$ by $(2\psi-1)T$ matrix of coefficients for vector Y_i in the conservation of flow constraints 5 and 6. The general structure of such a matrix is presented below:

$$N_i^y = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & -1 & \cdots & 1 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & -1 & -1 & \cdots & 1 & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & -1 & -1 & \cdots & 1 & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 \end{bmatrix} \quad \textbf{(40)}$$

The columns with the non zero coefficients in row 2, $t+1$, T and $T+1$ correspond to the variables $y_{i,1,1,0}$, $y_{i,1,2,0}$, $y_{i,1,1,1}$, $y_{i,1,2,1}$, $y_{i,j-1,j,t-1}$, $y_{i,j,j,t-1}$, $y_{i,j,j,t}$, $y_{i,j,j+1,t}$, $y_{i,\psi-2,\psi-1,T-2}$, $y_{i,\psi-1,\psi-1,T-2}$, $y_{i,\psi-1,\psi-1,T-1}$, $y_{i,\psi-1,\psi,T-1}$ and $y_{i,\psi,\psi,T-1}$ respectively.

Matrix N_i^z is a $(\psi(T+1)+D+1)$ by DT matrix corresponding to the coefficients of variables z_{idl} in the conservation of flow constraints (sets 4, 5, 6 y 7).

$$N_i^z = \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \\ 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 1 & \cdots & 0 & \cdots & 1 \end{bmatrix} \quad \textbf{(41)}$$

The rows with non-zero elements are 2, j_1+1, j_D+1, $\psi(t-1)+j_1+1$, $\psi(t-1)+j_D+1$, $\psi(T-1)+j_1+1$ $\psi(T-1)+j_D+1$, where sub index j_d are the nodes in the pipeline from which it is possible to visit depot d, $d=1,\ldots,D$. The columns with non-zero elements are $z_{i,1,1}$, $z_{i,D,1}$, $z_{i,1,t}$, $z_{i,D,t}$, $z_{i,1,T}$, $z_{i,D,T}$. The right hand side vector for the set of conservation of flow constraints for product i is given by:

$$RHS_i = [1 \quad \rho_{i,1,0} \quad \cdots \quad \rho_{i,\psi,0} \quad 0 \quad \cdots \quad 0 \quad \rho_{i,1,T} \quad \cdots \quad \rho_{i,\psi,T} \quad R_{i1} \quad \cdots \quad R_{iD}]^T \quad (42)$$

APPENDIX 2

The coefficients for the components of vector $\vec{y}_{il}$ in this set of constraints are given by the matrix G_{il} that is shown below:

$$G_{il} = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & 1 & 1 \end{bmatrix} \quad \textbf{(43)}$$

This corresponds to vector in equation (16). The coefficients for the components of vector X_i in this set of constraints are given by matrix G_i^X defined as follows:

$$G_i^X = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \quad \textbf{(44)}$$

The rows with a nonzero element under each column are given by: $j = 1 + l\psi$, where $l = 1, \cdots, T$
The coefficients of vector Y_i in this set of constraints are given by

$$G_i Y_i = \begin{bmatrix} G_{i0} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & G_{il} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & G_{iT-1} \end{bmatrix} \begin{bmatrix} \vec{y}_{i0} \\ \vdots \\ \vec{y}_{il} \\ \vdots \\ \vec{y}_{iT-1} \end{bmatrix} \quad \textbf{(45)}$$

So then the set of constraints in matrix notation is expressed as:
Both, matrix G_i^X and G_i have ψT rows.

APPENDIX 3

Let's consider now the structure of set of constraints (28): Let $T_{Y_{il}}$ to be the matrix of coefficients for variables $\vec{y}_{il}$. $T_{Y_{il}}$ is a $(\psi-1)$ by $(2\psi-1)$.

$$T_{Y_{il}} = \begin{bmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 0 \end{bmatrix} \quad \textbf{(46)}$$

Let T_i^1 to be the matrix defined as:

$$T_i^1 = \begin{bmatrix} T_{Y_{i0}} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & T_{Y_{il}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & T_{Y_{iT-1}} \end{bmatrix} \quad \textbf{(47)}$$

Such a matrix has $T(\psi-1)$ rows and $T(2\psi-1)$ columns. Analogously, for product q we define the corresponding two matrices:

$$T_{Y_{ql}} = \begin{bmatrix} 0 & 0 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(48)}$$

Let T_q^2 to be the matrix defined as:

$$T_q^2 = \begin{bmatrix} T_{Y_{q0}} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & T_{Y_{ql}} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & T_{Y_{qT-1}} \end{bmatrix} \quad \textbf{(49)}$$

with the same dimensions as for the case of product i. This is valid for any pair of 2 products (i,q). For the coefficients of the Z variables involved in this set of constraints, let's consider the following definitions: first let

$$\delta\nu = \{(j,d) : (CS_1,1),\cdots,(CS_d,d),\cdots,(CS_{D-1},D-1)\} \qquad \textbf{(50)}$$

to be the set of pairs node-depot where the first component of the pairs corresponds to the sub index of the last node of a given segment and the second element of the pairs corresponds to the sub index identifying the respective depot. The form of matrix T_{zl} is given as:

$$T_{zl} = \begin{bmatrix} 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{bmatrix} \qquad \textbf{(51)}$$

The non-zero elements appear in the coordinates corresponding to the pairs $(j,D) \in \delta\nu$. This matrix has $\psi - 1$ rows and D columns

Let T_z^3 to be the matrix defined as

$$T_z^3 = \begin{bmatrix} T_{z1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & T_{zl} & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & T_{zT} \end{bmatrix} \qquad \textbf{(52)}$$

That is a $T(\psi - 1)$ by TD matrix.

So then for every pair of products $(i,q), i \neq q$, the general form for the transition constraints is given as follows:

$$\begin{array}{ccccccc} T_{Y_{i0}}Y_{i0} + T_{Y_{q0}}Y_{q0} + T_{z1}z_1 - 2Iu_{iq1} & \cdots & 0 & \cdots & 0 & \geq & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & T_{Y_{il}}Y_{il} + T_{Y_{ql}}Y_{ql} + T_{zl+1}z_{l+1} - 2Iu_{iql} & \cdots & 0 & \geq & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & \cdots & T_{Y_{iT-1}}Y_{iT-1} + T_{Y_{qT-1}}Y_{qT-1} + T_{zT}z_T - 2Iu_{iqT} & \geq & 0 \end{array}$$

Figure 24: Structure of the additional constraints when interfaces are considered

Where I corresponds to an identity matrix of dimension $\psi - 1$ by $\psi - 1$. The total number of constraints for the pair of products (i,q) is $T(\psi - 1)$ and the total number of transition constraints is $P^2 T(\psi - 1)$

VITA

Rolando José Acosta Amado was born in Cúcuta, Colombia and raised in Suaita, a small municipality located in the department of Santander in Colombia, South America. He graduated from Lucas Caballero High School at Suaita and then moved to Bucaramanga, the capital city of the Department of Santander and enrolled in the Industrial Engineering undergraduate program of the Industrial University of Santander (in Spanish: Universidad Industrial de Santander-UIS). He received his undergraduate degree in Industrial Engineering in year 2000. In 2001, he enrolled in the master's program in Industrial Engineering in the University of Los Andes at Bogota, Colombia. He received his master's degree in Industrial Engineering in 2003. Between years 2002 and 2006, he worked as an assistant faculty member in the Department of Industrial Engineering of the Bolivarian Pontifical University at Bucaramanga and as an adjunct assistant professor in the department of Industrial Engineering of the Industrial University of Santander at Bucaramanga. His current position is assistant professor in the Department of Industrial Engineering at the Bolivarian Pontifical University in Bucaramanga. Rolando is married to Ivonne and their first baby is on its way. He is son of Jorge and Ilvia and brother of Jorge, Ricardo and Roy. He also has 2 nephews: Jorge Arquimedes and Ilvia Maria.