IJISSCM Editorial Board

Editor-in-Chief: John Wang, Montclair State U., USA

Associate Editors:
Indranil Bose, The U. of Hong Kong, Hong Kong
Svein Bråthen, Molde U. College/Molde Research Institute, Norway
Steve Buckley, IBM TJ Watson Research Center, USA
Metin Cakanyildirim, U. of Texas at Dallas, USA
Charu Chandra, U. of Michigan, Dearborn, USA
K. Ganesh, McKinsey & Company, India
Hans-Henrik Hvolby, Aalborg U., Denmark
Paul D. Larson, U. of Manitoba, Canada
Dirk Christian Mattfeld, Technische Universität Braunschweig, Germany
Shams Rahman, RMIT U., Australia
Ing. Fouad Riane, Catholic U. of Mons, Belgium
Jan Riezebos, U. of Groningen, The Netherlands
Robert Trent, Lehigh U., USA
Dharmaraj Veeramani, U. of Wisconsin-Madison, USA
Abbott Weiss, MIT, USA
Dan Zhu, Iowa State U., USA

IGI Editorial:
Heather A. Probst, Senior Editorial Director
Jamie M. Wilson, Assistant Director of Journal Publications
Chris Hrobak, Journal Production Manager
Gregory Snader, Production and Graphics Assistant

International Editorial Review Board:

Robert Aboolian, California State U. San Marcos, USA
Antony Arokia Durai Raj K, Infosys Technologies Limited, India
Jayal Ashayeri, Tilburg U., The Netherlands
Ozden Bayazit, Central Washington U.-Lynnwood, USA
Laird Burns, U. of Alabama in Huntsville, USA
Ruth Carrasco-Gallego, Universidad Politécnica de Madrid, Spain
Frank (Youhua) Chen, The Chinese U. of Hong Kong, Hong Kong
Si Chen, Murray State U., USA
Richard Cho, U. of New Brunswick in Saint John, Canada
Tsan-Ming Choi, The Hong Kong Polytechnic U., China
Shoshanah Cohen, Stanford U., USA
Melanie De Grano, IBM, USA
S G Deshmukh, Indian Institute of Technology, India
Mustafa K. Dogru, Bell Labs, Ireland
Matt Drake, Duquesne U., USA
Ephrem Eyob, Virginia State U., USA
Luiz Felipe, Pontificia Universidade Católica do Rio de Janeiro, Brazil
Tianjun Feng, U. of California Irvine, USA
Xiuzhen (Susan) Feng, Beijing U. of Technology, China
Peter Fettke, Institute for Information Systems, Germany
Petr Fiala, U. of Economics Prague, Czech Republic
Robert Frankel, U. of North Florida, USA
Jean-Marc Frayret, Frayret École Polytechnique of Montréal, Canada
Jan Frick, U. of Stavanger, Norway
Saharidis Georgios, Rutgers U., USA
Alain Halley, HEC Montréal, Canada
J. Michael Hardin, U. of Alabama, USA
Kimberly Killmer Hollister, Montclair State U., USA
Takamichi Hosoda, Cardiff U., UK
Jeffrey Hsu, Fairleigh Dickinson U., USA
Kai Huang, Binghamton U., USA
Vipul Jain, Indian Institute of Technology Delhi, India
Wooseung Jang, U. of Missouri – Columbia, USA
Sanjay Jharkharia, Indian Institute of Management Kozhikode, India
Mingzhou Jin, Mississippi State U., USA
Minjoon Jun, New Mexico State U., USA
Damir Kalpic, U. of Zagreb, Croatia
Ibrahim Al Kattan, American U. of Sharjah, UAE
Ronald Klimberg, Saint Joseph's U., USA
Canan Kocabasoglu-Hillmer, U. of Kansas, USA
Vivekanand Kochikar, Infosys Technologies, India
Amarpreet S. Kohli, U. of Southern Maine, USA
Ching-Chung Kuo, U. of North Texas, USA
Ik-Whan G. Kwon, St. Louis U., USA
Hong Tau Lee, National Chin-Yi U. of Technology, Taiwan
Hsun-Ming (Sam) Lee, Texas State U.- San Marcos, USA
Helene Lidestam, Linköping U., Sweden
Pitipong Lin, IBM Global Supply Advanced Analytics Center of Excellence, USA
Huafan Ma, Indiana U., USA
Lijun Ma, The Chinese U. of Hong Kong, Hong Kong
Debasish N. Mallick, U. of St. Thomas, USA
Matthew Manary, Intel Corporation, USA
Sathasivam Mathiyalakan, Winston State Salem U., USA
Ajay Mishra, SUNY Binghamton, USA
Subhro Mitra, U. of North Texas at Dallas, USA
Charles Møller, Aalborg U., Denmark
Seong-Hyun Nam, U. of North Dakota, USA
Brian Neureuther, SUNY Plattsburgh, USA
Peter O'Neill, Monash U., Australia
Ulas Ozen, Bell Labs Ireland, Ireland
Rao Panchalavarapu, Schneider National Inc., USA
Greg Parlier, Institute for Defense Analyses (IDA) & Independent Consultant, USA
Praveen Pathak, U. of Florida, USA
Vittal Prabhu, Penn State U., USA
Nyoman Pujawan, Sepuluh Nopember Institute of Technology, Indonesia
Ellen Bolman Pullins, U. of Toledo, USA
Zhaoqiong Qin, Embry-Riddle Aeronautical U., USA
Carl M. Rebman, U. of San Diego, USA
Changrui Ren, IBM China Research Lab, China
Pedro M. Reyes, Baylor U., USA
Reza Saen, Islamic Azad U.-Karaj Branch, Iran
Rezzan Savaskan-Ebert, Northwestern U., USA
Annibal Scavarda, Brigham Young U., USA
Dogan Serel, Bilkent U., Turkey
Min Shi, California State U. - Los Angeles, USA
Junho Shim, Sookmyung Women's U., Korea
Judy Siguaw, Cornell-Nanyang Institute, Singapore
ManMohan Sodhi, City U. London, UK
Yuyue (Peter) Song, Memorial U. of Newfoundland, Canada
Rajagopalan Srinivasan, National U. of Singapore, Singapore
Ping Su, SUNY at New Paltz, USA
Albert Wee Kwan Tan, Malaysia Institute of Supply Chain Innovation (MIT SCALE), Malaysia
Marietta Tretter, Texas A&M, USA
Pitipong Veerakamolmal, IBM Global Services, USA
Liya Wang, U. Affiliated Research Center, USA
Yunzeng Wang, U. of California, USA
Chuni Wu, Hsing-Kuo U. of Management, Taiwan
Nan Xia, U. of Wisconsin-Milwaukee, USA
Yu Xia, Middle Tennessee State U., USA
Tiaojun Xiao, Nanjing U., China
Ruiliang Yan, Virginia State U., USA
Kyung Hoon Yang, U. of Wisconsin-La Crosse, USA
Junfang Yu, Southern Methodist U., USA
Amy Z. Zeng, Worcester Polytechnic Institute, USA
Ding Zhang, State U. of New York, USA
Jiang Zhang, Adelphi U., USA
Fang Zhao, American U. of Sharjah, United Arab Emirates
Bin Zhou, Kean U., USA
Honggeng Zhou, U. of New Hampshire, USA
Xiaowei Zhu, West Chester U., USA

IGIP
IGI PUBLISHING
WWW.IGI-GLOBAL.COM

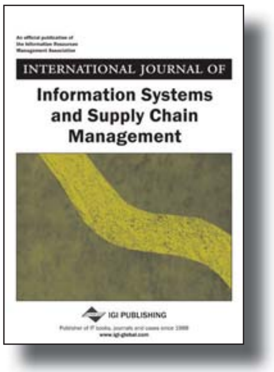
CALL FOR ARTICLES

International Journal of Information Systems and Supply Chain Management

An official publication of the Information Resources Management Association

The Editor-in-Chief of the *International Journal of Information Systems and Supply Chain Management* (IJISSCM) would like to invite you to consider submitting a manuscript for inclusion in this scholarly journal.

The *International Journal of Information Systems and Supply Chain Management* (IJISSCM) provides a practical and comprehensive forum for exchanging novel research ideas or down-to-earth practices which bridge the latest information technology and supply chain management. IJISSCM encourages submissions on how various information systems improve supply chain management, as well as how the advancement of supply chain management tools affects the information systems growth. The aim of this journal is to bring together the expertise of people who have worked with supply chain management across the world for people in the field of information systems.



ISSN 1935-5726
eISSN 1935-5734
Published quarterly

COVERAGE:

- Advanced planning and scheduling
- Artificial intelligence and decision making
- Bias, misinformation, and disinformation in informing systems
- Business intelligence and knowledge management
- Case-based reasoning
- Component-based information systems
- Computational econometrics
- Conceptual modeling
- Creative decision making and problem solving
- Customer value and customer relationship management
- Decision-making systems
- Decision making on supply chain management
- Decision models for information system design
- Decision-support systems for supply chain management
- Demand management
- E-business and decision making
- E-commerce and supply chain management
- Enterprise information systems
- Expert systems
- Geographic information systems
- Group support systems and groupware technologies
- Hybrid intelligent systems
- Implementing and integrating supply chain management
- Information systems and economic analysis
- Information systems and organizational behavior
- Information systems security
- Intelligent Systems over the Internet
- Interfirm networks
- IT for business forecasting
- Just-in-time and lean systems
- Logistics management
- Logistics network configuration
- Matching supply with demand
- Management Information Systems
- Modeling and analysis
- Multimedia and decision making
- Multiple-objective decision making
- Optimization and simulation
- Organizational information systems
- Performance management
- Reasoning and decision making under uncertainty
- Revenue management
- Risk management
- Strategic information systems and strategic alliances
- Supplier and customer relationship management
- Supply chain integration and coordination

All submissions should be e-mailed to:
John Wang, PhD
j.john.wang@gmail.com; journalswang@gmail.com

Ideas for Special Theme Issues may be submitted to the Editor-in-Chief.

Please recommend this publication to your librarian. For a convenient easy-to-use library recommendation form, please visit: http://www.igi-global.com/ijisscm

INTERNATIONAL JOURNAL OF INFORMATION SYSTEMS AND SUPPLY CHAIN MANAGEMENT

January-March 2012, Vol. 5, No. 1

Table of Contents

RESEARCH ARTICLES

Managing Distribution in Refined Products Pipelines Using Discrete-Event Simulation

M. Fernanda Gleizes, INTEC (CONICET-UNL), Argentina

Germán Herrero, INTEC (CONICET-UNL), Argentina

Diego C. Cafaro, INTEC (CONICET-UNL), Argentina

Carlos A. Méndez, INTEC (CONICET-UNL), Argentina

Jaime Cerdá, INTEC (CONICET-UNL), Argentina

ABSTRACT

The management of oil-product pipelines represents a critical task in the daily operation of petroleum supply chains. Efficient computational tools are needed to schedule pipeline operations in a reliable and cost-effective manner. This work presents a novel discrete event simulation system for the detailed scheduling of a multi-product pipeline consisting of a sequence of pipes that connects a single input station to several receiving terminals. The pipeline is modeled as a non-traditional multi-server queuing system involving a number of servers at every pipe end that perform their tasks in a synchronized manner. By using alternative priority rules, the model decides which server should dispatch the entity waiting for service to the associated depot. Also, the model deals with the timely fulfillment of terminal demands and the system response to unexpected events. In combination with optimization tools, the proposed simulation technique permits to easily manage real-world pipelines operations with low computational effort.

Keywords: *Delivery Schedule, Discrete-Event Simulation, Multi-Server Queuing System, Petroleum Supply Chain, Pipeline Operations*

INTRODUCTION

Pipeline management plays a key role in the petroleum business. Refined products pipelines are regarded as the most reliable and cost-efficient way to transport large amounts of liquid fuels over long distances. In contrast to other transportation modes, pipelines can operate continuously with almost no interruptions despite bad weather conditions. Furthermore, they have an important edge on environmental and safety issues. Pipelines transport a variety of oil derivatives in successive batches and operate in two different ways: segregated or fungible mode. Segregated products are branded or blend stock materials destined for

DOI: 10.4018/jisscm.2012010104

some specific client so that the same batch that is received for shipment in the origin is delivered to the client. Moreover, fungible batches consist of generic products fulfilling standard specifications. Shippers will receive a batch containing an equivalent product featuring the same specifications, but may not be the original lot they provide at the origin.

Pipeline Scheduling

The pipeline scheduling process is aimed at developing both the input and the delivery schedules. On one hand, the input schedule indicates the sequence of pump runs at every input terminal, as well as the injected product, batch size, starting time and pump rate for each run. Finding the optimal product sequence and lot sizes is a combinatorial problem that seeks to minimize interface costs due to product mixing while satisfying promised delivery dates. Since separation devices are rarely used, some sequences are definitely forbidden because of product contamination. On the other hand, the delivery schedule specifies the product batches partially/totally leaving the pipeline and the amounts diverted to the assigned destinations on every pumping run. In addition, it provides the times at which pumps should be turned on/off and valves at terminals are to be open/close for accomplishing the delivery plan. Its main goal is to meet depot demands while lowering the number of pipeline stoppages and pump switching to get savings on both energy costs for restarting flow in idle segments, and pump maintenance costs. Most of the computational burden in pipeline scheduling comes from these three difficult tasks: pump sequencing, batch sizing, and batch allocation to receiving terminals. By heuristically choosing them, the remaining operational decisions can be taken in a short CPU time. However, the final pipeline schedule is greatly influenced by those heuristic-based decisions previously taken (Boschetto et al., 2008).

Different approaches were proposed to study pipeline scheduling problems, including rigorous optimization models, knowledge-based techniques (Sasikumar et al., 1997), discrete-event simulation (Mori et al., 2007; García-Sánchez et al., 2008), and decomposition frameworks (Hane & Ratliff, 1995; Neves et al., 2007; Boschetto et al., 2008; Moura et al., 2008). Rigorous optimization methods generally consist of a single MILP (Mixed Integer Linear Programming) or MINLP (Mixed Integer Nonlinear Programming) mathematical formulation and are usually grouped into two classes: discrete and continuous, depending on the way volume and time domains are handled. Discrete formulations divide both the pipeline volume into a number of single-product packs, and the planning horizon into multiple time intervals (Magatão et al., 2004; Zyngier & Kelly, 2009; Rejowski & Pinto, 2003, 2008; Herrán et al., 2010). Most of them generally use uniform time and volume discretization. However, a later paper of Rejowski and Pinto (2008) assumes that each pipeline segment is composed by packs with equal or different pre-specified volumes to account for variations in the pipeline diameter, and the scheduling horizon comprises time intervals of adjustable duration to allow changes in the pump injection rate. On the other hand, available MILP-continuous optimization tools for pipeline scheduling, like the one proposed by Cafaro and Cerdá (2004, 2008), do not require any kind of decomposition or discretization scheme. They are able to find the optimal schedule for a single-origin pipeline with multiple output terminals by minimizing the sum of pumping, interface and inventory costs. Recently, the same authors developed an MILP-continuous formulation for scheduling pipeline networks with multiple input terminals (Cafaro & Cerdá, 2009). In any case, however, continuous approaches just provide the set of "aggregate" batch stripping operations to be done during every pumping run without specifying the detailed sequence of batch "cuts" to be performed by the pipeline operator. Generally, there are many ways to disaggregate such accumulated product discharges, i.e., several alternative cut sequences. Hence, the development of an efficient detailed output schedule through a continuous optimization approach is a difficult task that has not been solved yet. A complementary tool like discrete event simulation is then needed to convert the “aggregate” output planning provided by the optimization module into a sequence of individual cut operations to be executed during a batch injection.

In the last years, several hybrid approaches decomposing the pipeline scheduling process into a number of stages and applying different techniques to solve them have been published. A hybrid approach that combines a randomized constructive heuristic with novel constraint programming (CP) models was reported by Moura et al. (2008). It comprises two phases. The planning phase uses heuristics to create the set of batches to be injected (delivery orders), by specifying their volume, origin, destination depot, product type, assigned route, and delivery due date. The scheduling phase takes such a set of delivery orders and determines the sequence and start times of pumping operations to perform at every source node to meet the promised delivery dates. This second phase was implemented through a pair of CP models. Neves et al. (2007) presented another decomposition approach for the planning of pipeline operations over a monthly horizon. The decomposition relied on a heuristic-based pre-processing block that accounts for demand requirements, production planning at oil refineries, and typical lot sizes to determine a candidate set of product sequences. In addition, the heuristic block provides time windows for pump and delivery operations at every terminal. Afterwards, the pre-processed information is used by a continuous-time MILP formulation to determine the exact start/finish times of batch input and delivery operations. Since seasonal costs of the electric energy are considered, the model includes binary variables just to avoid pumping operations during high-energy cost periods. Boschetto et al. (2008) reformulated the hybrid approach of Neves et al. (2007) using a different decomposition strategy now involving three blocks: (1) a resource-allocation block determining candidate sequences of batch injections, (2) a pre-analysis block specifying the precise volumes to be either pumped from source nodes or received in destination nodes, and providing the earliest start/finish times for stripping operations in every destination node, and (3) an MILP model determining the exact timing of pump and delivery operations at each node. In contrast to continuous-time formulations, approximate hybrid approaches provide very detailed pipeline output schedules.

Simulating Pipeline Operations

In the last decade, few contributions on discrete event simulation of refined products pipeline operations have been published. Mori et al. (2007) developed a simulation model for the scheduling of injection and stripping operations in a real-world pipeline network. The network consists of a series of single pipes that connect multiple refineries, harbors and distribution centers among themselves, and transport many oil derivatives. García-Sánchez et al. (2008) presented a hybrid methodology combining tabu search and a discrete-event simulation model for addressing a real-world pipeline scheduling problem. In this work, tabu search technique is used to improve non-optimal pipeline schedules that are subsequently tested by the simulation model. Product batches are divided into equal-sized discrete packs whose destinations are pre-defined at the time they are injected into the pipeline network.

Our work introduces a discrete event simulation model for a trunk pipeline transporting refined products from a single origin to multiple distribution terminals on segregated or fungible mode. The trunk line is made up of a sequence of pipes, each one connecting either an input to an output terminal or just a pair of distribution terminals between themselves. The detailed pipeline injection schedule to be simulated is provided by an optimization module that implements the continuous-time mathematical formulation of Cafaro and Cerdá (2004, 2008). From the discrete simulation viewpoint, the pipeline can be regarded as a multi-server queuing system. Those servers perform their tasks in a synchronized manner, with each one having its own queue of fixed-sized batch elements, also called entities. There is a server at the end of each pipe and its queue is composed by the sequence of batch elements contained in that pipe. Since every pipeline segment should be permanently full of liquid and has a constant volume, the length of any server queue will remain fixed throughout the whole time horizon. At every pumping run, there is a limited set of servers that are eligible to dispatch an elementary batch to the related terminal. Such a set of eligible servers is inferred from the aggregate delivery schedule provided by the optimization module. At most, only one server can be active at a time. To choose the active server dispatching an entity at every pumping event, alternative heuristic rules have been proposed. The simulation model also includes some constraints forcing to meet the aggregate product deliveries to terminals prescribed by the pipeline planning at every run. In this way, detailed input and output schedules are generated.

AN ILLUSTRATIVE EXAMPLE

Figure 1 shows a typical multiproduct pipeline transport system. It consists of a single refinery where oil products are injected, and five receiving terminals placed along the pipeline. The first line in Figure 1 depicts the location of every batch inside the pipeline (linefill) at the start of the time horizon. Initially, there are four batches of products *P2-P1-P3-P4* inside the pipeline, with 400-400-350-350 volumetric units respectively. The following line in Figure 1 illustrates the pipeline content after completing the first batch injection involving 400 units of product *P4* and running from 0.00 h to 8.00 h (represented by a right arrow at the origin). It can also be observed a series of up arrows indicating the aggregate products deliveries to every terminal taking place while injecting the batch of *P4*. Four product extractions from the line are accomplished: 100 units of *P2* to depot *D1*, 100 units of *P1* to *D2*, 100 units of *P3* to *D3*, 50 units of *P4* to *D4* and 50 units of *P4* to *D5*. Due to liquid incompressibility, the volume of the batch injected at the origin equals the sum of product deliveries to receiving depots, i.e., 400 = 100 + 100 + 100 + 50 + 50. The optimization model proposed by Cafaro and Cerdá (2004, 2008) not only generates an efficient input schedule but also provides the set of "aggregate" stripping volumes to be transferred to terminals during every batch injection. As it will be shown, there are many ways to accomplish the proposed non-detailed output schedule. For instance, one alternative is to prioritize product deliveries to the nearest terminal, as depicted in Figure 2.

On the other hand, it may occur that the pipeline operator is focused on fulfilling product demands at the farthest destination first. If

Figure 1. A typical input operation in a multiproduct pipeline system

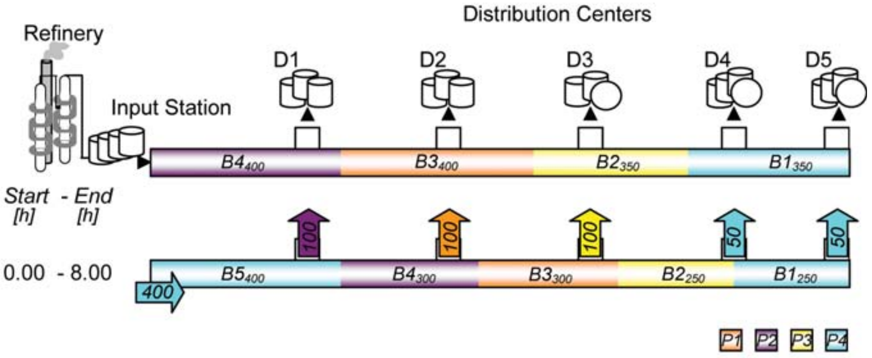
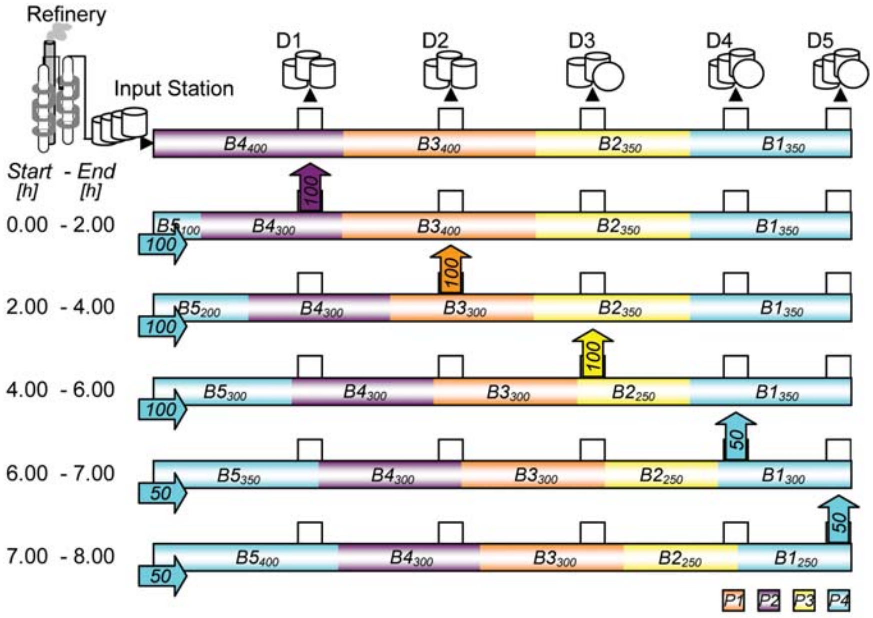


Figure 2. A detailed output schedule prioritizing product deliveries to the nearest terminal



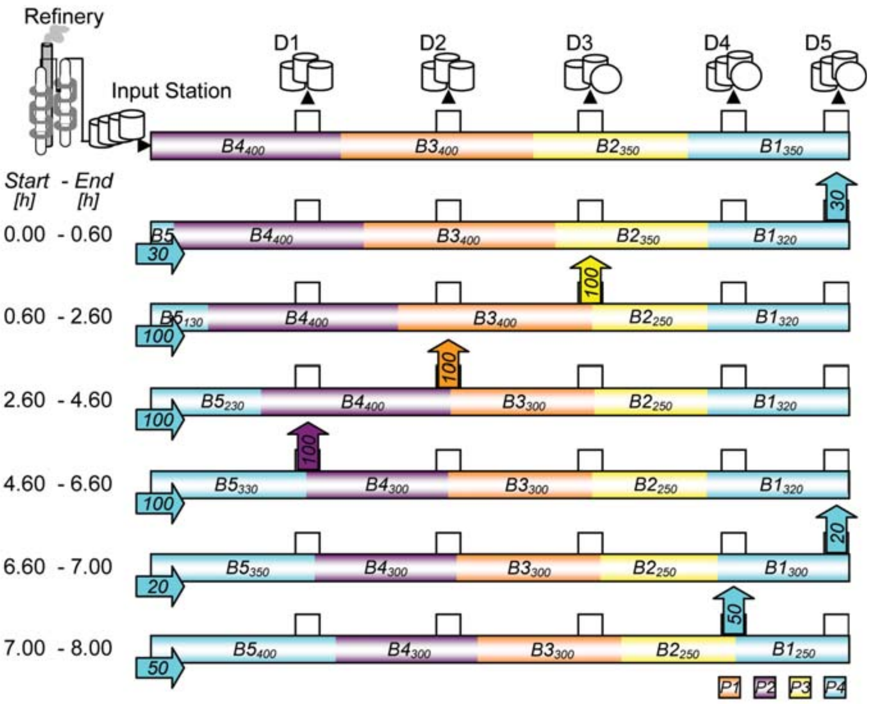
so, it results the detailed delivery schedule described in Figure 3. Not only the detailed output schedule changes, but also the sequence of shorter pumping runs ("short runs") injecting a total of 400 units of product *P4*.

Notice that some stripping operations to higher-priority terminals like *D5* (50 units) and *D4* (50 units) must be interrupted/postponed and later resumed/started to meet the aggregate deliveries shown in Figure 1. Otherwise, product deliveries to terminals *D3*, *D2* and *D1* become infeasible. Consequently, an "aggregate" stripping operation may be performed through two or more non-consecutive cuts on some batch all destined for the same terminal (see deliveries of product *P4* to terminal *D5* in lines 2 & 6 of Figure 3). Furthermore, the injection of 400 units of *P4* is performed through a sequence of six shorter pumping runs each one associated to a different cut operation. Two of them deliver product *P4* to depot *D5* from the batch located at the pipeline end. Such a detailed description of a batch injection is also needed by a pipeline operator and can be obtained through discrete event simulation. In the following sections we will discuss this matter with more detail.

A SIMULATION-BASED HIERARCHICAL SCHEDULING APPROACH

To develop a detailed delivery schedule like the ones presented in previous Section, the pipeline scheduling problem has been hierarchically solved in two steps with the first one generating the input schedule through the optimization module. At the second stage, a detailed output schedule based on the information provided by the optimization approach is developed. To do that, this work introduces an efficient discrete event simulation model developed on Arena® (Kelton et al., 2007) for both validating the pipeline schedule provided by the optimization module and generating the detailed output schedule. In addition, the proposed model

Figure 3. A detailed output schedule prioritizing product deliveries to the farthest terminal



permits to visualize the pipeline operations by means of a friendly animation interface showing the dynamics of the pipeline system over time.

The simulator uses as input data the set of batches to be stripped, the receiving depots and the related product amounts to be delivered to terminals while performing a pumping run. Such data are contained in the so-called terminal-batch delivery assignment matrix $Q_k^o(i,j)$ whose element $q_k^o(i,j)$ represents the aggregate volume of product deliveries from batch *i* to terminal *j* during pumping run *k*. For the batch injection illustrated in Figure 1, such an assignment matrix is given in Table 1. For instance, two "aggregate" volumes of batch *B1* containing *P4* are to be delivered during the injection of batch *B5*. The receiving terminals are *D4* and *D5*, and the amounts to supply to each depot are equal in both cases to 50 units.

However, the detailed order of execution of the individual stripping operations is not given. Besides, some of them could be done in two or more non-consecutive cuts. Then, some further work should be made before fully developing the pipeline delivery schedule. Such a task will be performed by the proposed simulation model. Since a pumping run is divided into a sequence of events, and batches flowing through the line are made up of entities, the possible destinations for an entity just to be served during run *k* can be easily derived from the current matrix $Q_k(i,j)$. When a new input event occurs, therefore, each pipe server knows if the first entity on its queue is eligible for being transferred to the associated terminal. Otherwise, the entity should either move to the next pipe or still remain at the same queue. If two or more servers can dispatch the leading entry to their output terminals in order to meet unsatisfied demands, the simulation model should decide, based on priority rules, which one is chosen. To simulate the pipeline schedule provided by the optimization package, the specified aggregate product deliveries for every

Table 1. Aggregate product deliveries to depots while injecting a batch of product P4

	D1	*D2*	*D3*	*D4*	*D5*
B1(P4)	0	0	0	50	50
B2(P3)	0	0	100	0	0
B3(P1)	0	100	0	0	0
B4(P2)	100	0	0	0	0
$B5(P4)_{new}$	0	0	0	0	0

run k given by $q_k^o(i,j)$ must be achieved. After every event, the simulation model should update the unsatisfied demands $q_k(i,j)$ in every terminal, initially equal to $q_k^o(i,j)$. If $q_k(i,j)$ drops to zero, such a terminal j is no longer eligible to receive an entity from batch i. When all unsatisfied demands $q_k(i,j)$ are null, the detailed output schedule for run k has been generated.

Previous contributions (Mori et al., 2007; García-Sánchez et al., 2008) assumed that the destination for each entity was already given by the optimization package. In our approach, some capabilities have been provided to the proposed simulation model for selecting the route to be followed by every entity based on three key elements: (1) the assignment matrix $Q_k^o(i,j)$ for every run k, (2) the batch to which each entity belongs, and (3) a set of priority rules selecting both the leaving entity and the receiving terminal, if several cut operations are eligible for execution. Different pipeline output schedules can indeed be generated by changing the priority rules. The simulation model can also consider operational details like loading and unloading materials of individual tanks at input and distribution terminals, instead of handling them in an aggregate manner. As a result, the simulation model can track the evolution of the inventory in every individual tank over time. Moreover, pipeline stoppages due to high-cost peak periods can also be handled.

PROBLEM STATEMENT

Given: (a) a multiproduct pipeline system connecting an oil refinery to several distribution terminals; (b) the number and type of products to be transported through the pipeline system; (c) the sequence of product batches to be introduced in the origin of the pipeline system (input schedule); (d) the associated set of stripping volumes to be transferred during every batch injection to the receiving terminals (i.e., the aggregate output schedule given by the terminal-batch assignment matrix $Q_k^o(i,j)$); (e) the scheduled production runs to be loaded into the tanks at the input station; (f) the initial pipeline conditions (sequence of batches inside the line at $t = 0$ and their sizes); (g) the initial stock of every product in terminal tanks; (h) the product demand profile at distribution centers; (1) the product pumping rate; and (j) the horizon length.

It should be established the detailed input and delivery schedules, including:

1. The detailed sequence of batch injections, each one comprising a series of shorter pumping operations (called runs in the rest of the paper);
2. The product and volume injected in the line during every run;
3. The starting/end time of each run;
4. The amount and type of product delivered, the batch source and the receiving terminal for every run;
5. The product inventory management at the input station by considering discharged production runs from neighboring refineries and product lots injected into the pipeline;
6. The product inventory management at receiving terminals by considering dis-

charged product lots and client demands on a hourly basis.

MAJOR ASSUMPTIONS

In order to ensure that the proposed simulation model to pipeline operation is a faithful representation of the real-world system, all inherent problem features are explicitly taken into account in the discrete event simulation framework. These essential characteristics are mostly related to the detailed modeling of the particular system topology and the actual operating conditions. Thus, the major assumptions that were considered to derive the proposed simulation framework are the following:

1. A unidirectional pipeline connecting a single refinery to multiple distribution terminals is considered.
2. The pipeline is always full of liquid products and operates either on segregated or fungible mode. On the later mode, a single batch can have many destinations.
3. Batches of products are injected into the pipe one after the other, with no physical barrier between them.
4. The interface or contamination loss between a particular pair of refined products is a known constant.
5. Due to liquid incompressibility, every time a batch element is injected, one and only one entity already in the line is simultaneously transferred from the pipeline to a single receiving terminal.
6. Every run is performed at a fixed flow rate.
7. Distribution centers are tank farms with dedicated storage units of known capacity for each product.
8. At most one terminal tank is connected to the pipeline at any time event, and the setup time for switching from one tank to another is negligible.
9. Refinery production schedules have been previously developed. Scheduled start/end times and rates of incoming product flows to storage tanks at the input station are problem data.
10. Daily client demands are given on an hourly basis.

Based on the previous list of assumptions, the simulation model is able to achieve a very pragmatic representation of the inherent characteristics of the real-world system under study. Thus, it can be easily used as a very effective and user-friendly computer aided tool to emulate, analyze, improve and manage the daily operation of complex refined products pipelines.

QUALITY MEASUREMENT OF PIPELINE SCHEDULES

The sequence in which product deliveries are accomplished has great impact on the pipeline system operational costs. In particular, pipeline stoppages are quite expensive (Hane & Ratliff, 1995) and should be avoided. A pipe stoppage occurs whenever a delivery at some terminal is interrupted and a different stripping operation starts at an upstream (nearest-to-refinery) point. This causes the interruption of the flow in the pipeline segment connecting the recently activated and deactivated depots, and the consequent shutdown of several pump stations. The main cost of a stoppage is associated with the lost of energy of the fluid momentum, since the stopped flow must be put in motion again when restarting product supplies to downstream destinations. In addition to the increased energy cost, the maintenance cost rises with the number of stoppages because the time between pump repairs strongly depends on the number of shutdowns. To measure the quality of the resulting detailed schedule, the so-called *accumulated idle volume* is defined. This measurement is computed by adding the idle pipe volumes throughout the complete horizon. The accumulated idle volume together with the total number of cut operations required to meet the specified terminal demands are the two performance measures used to compare alternative pipeline schedules.

PROPOSED SIMULATION MODEL STRUCTURE

A refined product pipeline network is primarily composed of an input station, a set of pipes and receiving terminals. Tank storage facilities, such as the input station and the receiving terminals, are connected through pipes. Once oil products arrive to the input station coming from neighboring refineries they are temporarily stored in tanks until the execution of injection runs. A run implies the injection of material from tanks of the input station into the pipeline system. The type of product injected, the batch size and the expected pumping rate for each run are specified by the optimization module. Each injection run introduces a new batch volume that will be pumped into the line from the input station. The sequence of injection runs determines the input schedule. When a run is performed, the material in the line moves forward due to the injection of new batch volumes at the input station. While doing so, another lot already in transit along the line is simultaneously transferred to a single receiving terminal.

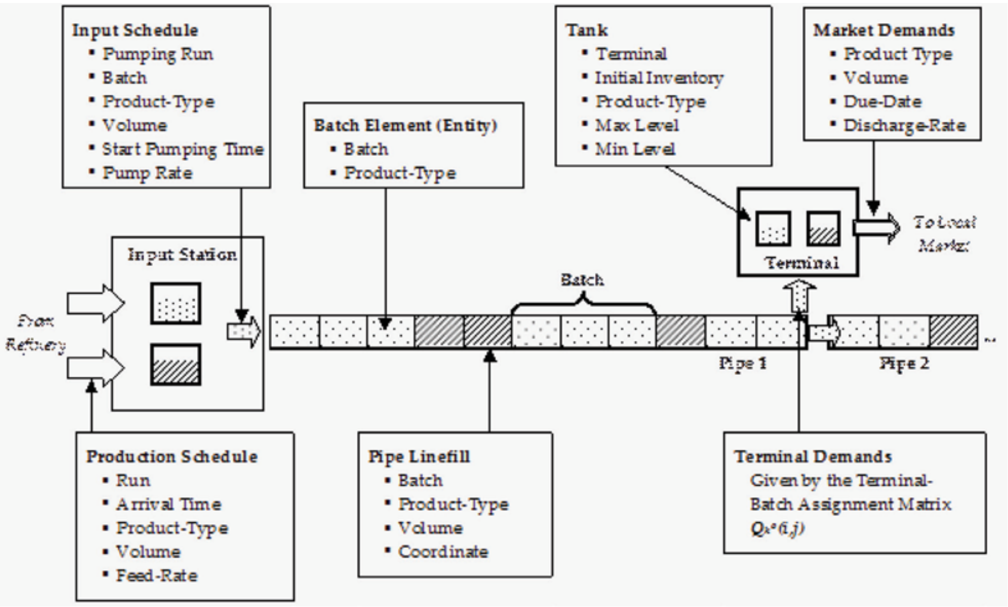
Model Structure

The simulation of pipeline operations is based on a list of time events, each one representing the injection start time of a single batch element into the system. The event list is generated by the scheduler block of Arena® based on the pipeline input schedule to be simulated. In our case, the input schedule provided by the optimization approach of Cafaro and Cerdá (2004, 2008) includes the sequence of batches to be injected and the batch attributes (product, volume, pump rate and start pumping time). Since product batches are handled in a discretized manner, their volumes are expressed as a number of small, equal-size batch elements called entities. Each entity contains a specific volume of a certain product. If the common entity size decreases, every batch injection will include a larger number of entities, and a more accurate model will be achieved at the expense of a higher computational cost. Hence, a major model decision is the selection of a proper entity volume. Entity attributes are inherited from the batch to which the entity belongs. Similarly, the linefill of a pipe consists of a sequence of entities defining the queue of the pipe server. Each event represents the pumping of a single entity into the line, and the state of the pipeline system just changes when a new event occurs.

Each time an elementary pumping operation is run at the input station (i.e., at the inlet of the first pipe) a new entity enters the first-server queue, and another entity at the exit of one of the pipes should be dispatched to the corresponding receiving terminal. The set of servers should jointly decide which one will dispatch the first entity on its queue to the associated distribution terminal (dispatching server). Moreover, they also establish which servers should transfer the accessible entity to the next pipe (servers located upstream of the dispatching server) and which ones will remain idle because there are no new arrivals (servers located downstream of the dispatching server).

In other words, a pipe server can take three different actions whenever an event is accomplished: (a) it remains idle because there is no product arrival to the queue; (b) it transfers the first entity waiting for service to the next pipe, or (c) it delivers the first entity on the queue to the receiving terminal. Actions (b) and (c) are triggered by a new entity arrival to the server queue. As a result, just a single distribution terminal will be active on every pumping event. The service time of an entity is defined in terms of two pieces of information: the entity volume (a user choice) and the pumping rate for the current operation. After the servers perform their jobs, the simulation clock is advanced to the next event time. Delays can arise if the selected distribution terminal has no enough storage capacity to receive the departure entity. Such pipeline "blockages" lead to the so-called disrupted pipeline operation.

Figure 4. Major components of the discrete-event simulation model



Simulation Blocks

The simulation model structure involves three blocks, each one representing a main component of the pipeline system: the input station, the receiving terminals, and the pipes. Figure 4 shows the model blocks together with key simulation elements, such as entities (batch elements) and resources.

Input Station

The input station is located at the origin of the pipeline system. Product batches are discharged from input tanks and injected into the line, while some production runs from the neighboring refineries are being loaded. The input station is composed of storage tanks and pumps, and the simulation system forces to stop the pipeline activity when the available stock is running out.

Operations at the input station are modeled based on two key data sets. The first one is the production schedule at the refinery. New products arrivals (production runs) are sent to storage tanks of the input station at pre-defined time events. The inventory level in every storage tank is modeled as a continuous variable, allowing to accurately controlling the evolution of product stocks at the input station. The second data set is the input schedule consisting of the sequence of batches to be injected into the line and the batch attributes: product type, volume, mean pump rate, and start pumping time.

Finally, a key decision to be made by this model component is the pump rate at which batch elements are to be injected. To do that, design and operational constraints should be taken into account. Pressure limits have to be respected, and turbulent flow should be maintained in order to minimize product interface volumes. A main term of the transportation cost function is associated to the energy consumed for moving product batches into the line. The pumping cost is strongly affected by the batch pump rate. This is the reason why the proposed simulation model is able to choose different pump rates for different runs, a critical matter to accurately represent the operation of a real pipeline system. A batch injection usually consists of a sequence of runs, and each run just produces a single batch cut operation.

Moreover, oil pipeline operators avoid running pump stations at daily peak periods because a much higher energy price must be paid for the electrical power consumption. The pipeline simulation system can easily account for high-pumping cost intervals.

Pipes

The pipeline system is divided into different segments (pipes) that connect input and output nodes. Each pipe is modeled as a fixed-size FIFO queue, with a single server at the pipe extreme that permits the movement of material entities from the pipe either to the associate depot or to the next conduct. By connecting pipes with different sizes and service rates, the model is able to simulate the operation of almost every pipeline system structure.

Every time an entity enters the pipeline segment at the inlet point it pushes the entity positioned at the other extreme out of the pipe. In other words, a server located at every pipe end dispatches the first entity on the queue either to the associated terminal or to the next pipe, whenever a new entity enters through the pipe inlet section. Since the system transports multiple products, the model keeps track of every pipe linefill by updating the server queues at every time event. The actions that a pipe server can take on the entity waiting for service are: (1) no action; (2) move to the next pipe; (3) load in a terminal tank.

At the beginning of the scheduling horizon it is necessary to create the initial condition of the pipeline content, which is basically carried out by the initialization process. The volume of each pipe together with the selected entity size determines the number of entities that the segment contains. This number remains constant all over the scheduling horizon, as the volume of each pipe is a fixed quantity given by the cross section and the segment length.

Terminals

Terminals are tank-farms from which products are sent to consumer markets. In the proposed simulation model, arrivals of product entities from the line and deliveries to regional markets are simultaneously handled.

Tanks have a maximum capacity that cannot be exceeded and, in general, a minimum level of material is also required. As previously mentioned, continuous variables are used to model the inventory level in storage tanks.

The pipeline output schedule is automatically generated by the simulation model taking into account the terminal demands (batch sources and aggregate stripped volumes) to be covered during every batch injection. Such terminal requirements are given by the terminal-batch assignment matrix, $Q_k^o(i,j)$, provided by the optimization module.

On the other hand, product batches are sent to local markets from terminal tanks mainly by truck. Market demands are continuously satisfied and tank inventories are updated accounting for the product lots received from the pipeline and the tank discharge rates. Thus, it is possible to define a daily or hourly pattern demand. Moreover, different probability distributions can be used to represent stochastic demands.

Priority Rules

When a batch entity reaches the position of a terminal, it may be transferred from the pipeline to an available tank. At every pumping event, only one terminal can receive a single entity from the line. Moreover, every material entity in the pipeline located between the input station and the selected terminal moves on, while the rest of the pipes remain idle. In order to decide which of the eligible servers should dispatch the first entity on its queue to the associated terminal, the simulation model uses a set of heuristic rules assigning different priorities to terminals, so as to choose the one that should receive the demanded batch entity first. For the sake of comparison, a pair of very simple priority rules was applied to solve the case study introduced in the next section:

(a) The *Nearest-First* (*NF*) rule prioritizes the product delivery to the eligible terminal closest to the origin. If two alternative terminals are eligible to receive a product unit, the one positioned closest to the input station will be selected. As a result, no required entity will overpass a demanding terminal since the most upstream location is always favored. In this way, the *NF* rule will always generate a feasible output schedule.

(b) The *Farthest-First* (*FF*) rule prioritizes product deliveries to the farthest eligible terminal from the origin. In this case, it is necessary to verify if the delivery of an entity to the prioritized terminal does not prevent from satisfying product demands at some upstream depot. Since a unidirectional pipeline is considered, an entity of batch i that overpasses its destination T_j during run k can no longer be transferred to it. If such an entity is absolutely necessary to meet the product demand of T_j covered by batch i during run k ($q_k(i,j)$) the resulting output schedule would be infeasible. In order to meet the requirement of such "restricted" depot T_j, the product delivery to the farthest eligible terminal should be interrupted to avoid that the entity of batch i overpasses T_j. In other words, no eligible depot located farther than T_j from the origin can be selected as the receiving terminal. Only an eligible terminal on the pipes connecting T_1 to T_j can be chosen to receive a batch entity. In that case, the farthest feasible terminal (T_j) is selected as the active terminal. If there are several eligible terminals that are restricted, the one closer to the origin T_j' is prioritized.

Simulation Algorithm

Each time an entity enters the pipeline system, the following six-step algorithm is carried out:

1. Identify the current eligible terminals: Let us assume that the current injection belongs to pumping run k, and the first entity on the server queue of terminal j comes from batch i. If the current value of $q_k(i,j)$ is greater than zero, then depot j is eligible to receive that entity of batch i.
2. Identify the eligible terminals that are restricted: Let us consider terminal j that is eligible to receive an entity of batch i. Compute the number of entities of batch i located between the origin of the pipeline system and the terminal j, $n(i,j)$. If $n(i,j)*v = q_k(i,j)$, then terminal j is restrictive. Parameter v stands for the common volume of every single entity.
3. Identify the most restricted eligible terminal. Among the restrictive eligible terminals, select the one T_j' closer to the origin.
4. Among the eligible terminals located between the pipeline origin and the most restricted terminal T_j', select the one T_j^* with the highest priority. Choose T_j^* as the receiving terminal.
5. Pump a new entity into the pipeline system and simultaneously transfer the entity of batch i waiting for service to the selected terminal T_j^*. Update $q_k(i,j)$ by reducing its value by the entity volume (v).
6. Move the first entity on the queue of any server closer to the origin (with regards to T_j^*) to the next pipe. Return to Step 1.

Therefore, the following pseudo-code can be defined.

By using different priority rules, the logic structure of the model generates alternative delivery schedules according to the current state of the pipeline and the depots requirements. The detailed delivery schedule is given by the sequence of batch entities transferred to receiving terminals over time.

CASE STUDY: A REAL-WORLD PIPELINE SCHEDULING PROBLEM

The proposed simulation-based model has been applied to the solution of a real-world problem first introduced by Rejowski and Pinto (2003) and later solved by Cafaro and Cerdá

Nomenclature.

```
SETS
I:          Ordered set of currently transported batches
J:          Ordered set of distribution terminals along the pipeline
E:          Subset of current eligible terminals
R:                 Subset of restricted terminals
PARAMETERS
n(i,j):current number of entities of batch i between the origin of the pipeline system and terminal j
q(i,j): current demand of batch i in terminal j (in volume units)
v:        entity volume (in volume units)
(1) E:= ∅
Forall j ∈ J
    Identify the accessible batch i*(j)
    If q(i*,j) > 0
        E := E ∪ {j}
    endif
endforall
(2) R:= ∅
Forall j ∈ E
    Compute n(i*,j)
    If n(i*,j)*v = q(i*,j)
        R := R ∪ {j}
    endif
endforall
(3) If R = ∅
    j':= J (the farthest destination)
else
    Identify j' ∈ R / j' is the closest-to-origin terminal in R
endif
(4) Identify j* ∈ J / j* is the terminal with the highest priority located between the pipeline origin and terminal j'
(5) Transfer an entity of batch i*(j*) to terminal j*
    q(i*,j*):= q(i*,j*) - v
(6) Return to (1)
```

(2004, 2008). This challenging scheduling problem addresses the daily operation of the OSBRA pipeline owned by PETROBRAS, which transports almost 20% of the total Brazilian oil derivatives. OSBRA is the most important Brazilian pipeline, which connects the REPLAN refinery located in Paulinia (Sao Paulo) with five distribution centers situated in the cities of Ribeirao Preto (Sao Paulo), Uberaba (Minas Gerais), Uberlandia (Minas Gerais), Goiania (Goias) and Brasilia (Distrito Federal). The pipeline system transports four refined products (gasoline, diesel, LPG and jet-fuel) to five distribution terminals along 955 km. Figure 5 shows the whole pipeline distribution network. The model makes use of the monthly input schedule generated by the continuous optimization model of Cafaro and Cerdá (2008). The pipeline system is divided into relatively small entities whose volume is fixed at 100 m^3. Thus, a total of 1635 entities are required to fill the whole line.

Figure 5. Distribution network of the OSBRA Brazilian pipeline

The Proposed Animated Interface

Figure 6 depicts the simulation-based interface developed on the Arena® simulation package. The animated interface allows the visualization of pipeline operations and the dynamic evolution of the pipeline network state. The software Arena® permits the development of graphic representations to assess the model operation over time. The main components of the pipeline system, i.e., the trunk pipeline, the input station and the receiving terminals, are also depicted in Figure 6. The current "status" of all pipeline segments is given by the product entities moving along the different queues. Each entity represents a fixed volume of a certain product, located in a specific place of the pipeline network at a given time event. Through this animation interface it is easy to follow the evolution of the pipeline content during the scheduling horizon. In addition, the inventory levels in storage tanks at the input station and receiving terminals are also traced by the model animation interface. As seen in Figure 6, two arrows, one at the input station and the other at the active receiving terminal, show the new batch being injected (*B12*) and the in-transit batch being stripped (*B11*), respectively. Moreover, global model variables like the objective function (accumulative idle volume) and still unsatisfied terminal demands $q_k(i,j)$ can easily be plotted in the animation interface, enhancing the comprehension of the model dynamics.

Applying the Proposed Simulation-Based System

Alternative delivery schedules for a monthly horizon can easily be generated and tested in less than one minute of CPU time by using the proposed simulation-based model with different priority rules.

Nearest-First (NF) Priority Rule: First Week Delivery Schedule

Figures 7 and 8 together with Table 2 represent the delivery schedule generated by using the *NF* rule for the first week of the monthly horizon. From $t = 5$ h to $t = 15$ h, a strip of batch *B4* is transferred to the eligible terminal *T2* closest to the origin, while the first run of *B6* is accomplished at the input station. At $t = 15$ h, the planned delivery is completed, and *B4* is now stripped to the farther terminal *T3*. Pipe 2-3 is activated, and the other pipes 0-1 and 1-2 between the origin and *T3* remain active during this second "cut" operation. At time $t = 21.67$ h, pipe 3-4 is also activated and *T4* begins to receive diesel fuel from batch *B2*. Almost 9 hours later ($t = 30.56$ h), the pipe 4-5 is set in motion and *T5* begins to receive gasoline from batch *B1*. Once the transfer of *B1* is completed ($t = 43.89$ h) the first pipe

Figure 6. View of the model animation interface

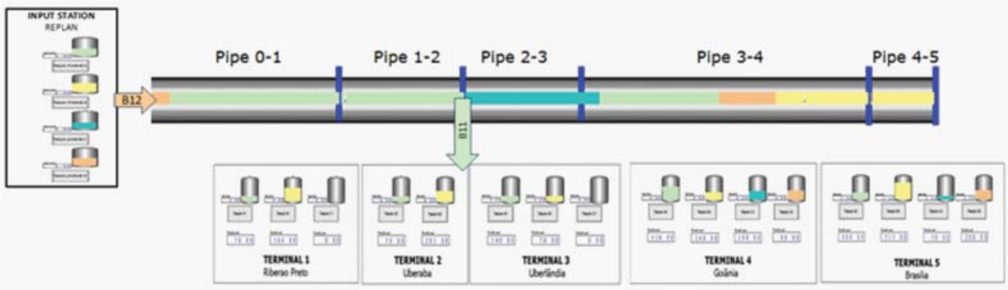
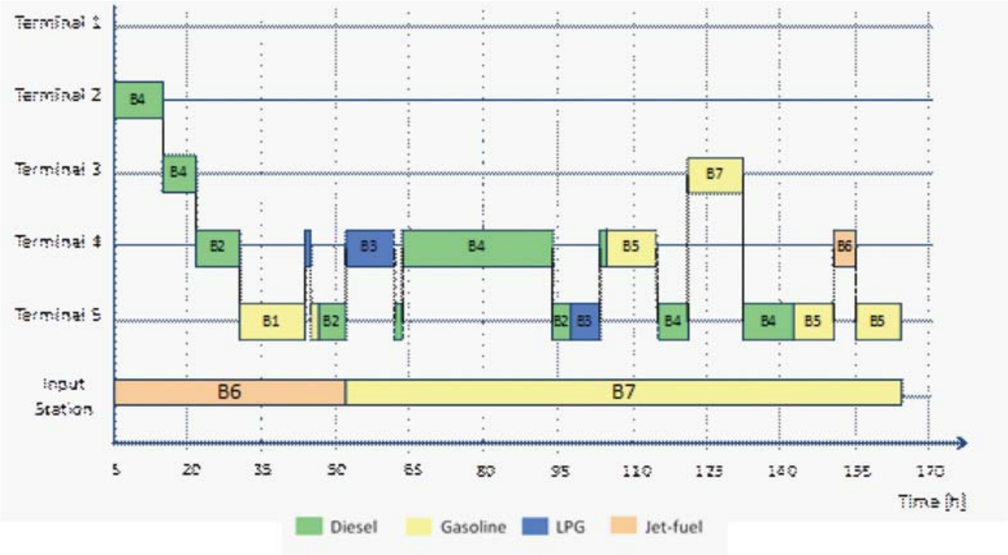


Figure 7. Delivery schedule using the NF-rule for the first week of the monthly horizon



stoppage occurs. Terminal *T4* begins to receive LPG from batch *B3* (Operation 5 in Figure 8) and pipe 4-5 is deactivated. The liquid content in that pipeline segment remains still and the accumulated idle volume is increased by 13,500 m^3. Next, the delivery of gasoline from *B1* to terminal *T5* is restarted to meet the aggregate allocation of *B1* to terminal *T5* prescribed by the element $q^o_{B6}(B1,T5)$ of the terminal-batch assignment matrix.

The accumulated idle volume is the result of summing all the stopped pipe volumes over the scheduling horizon. At the right side of Figure 8, it is shown the evolution of the accumulated idle volume at every delivery operation. Using the *NF* rule, 20 cut operations are required during the first week, and the overall accumulated idle volume amounts 141,000 m^3.

Farthest-First (FF) Priority Rule: First Week Delivery Schedule

Similarly to the *NF* rule, Figures 9 and 10 together with Table 3 describe the delivery schedule given by the *FF* rule for the first week of the planning horizon. Using the *FF* rule, 20 cut operations are also needed. However the

Figure 8. Active pipeline segments and evolution of the accumulated idle volume for the first week using the NF-rule

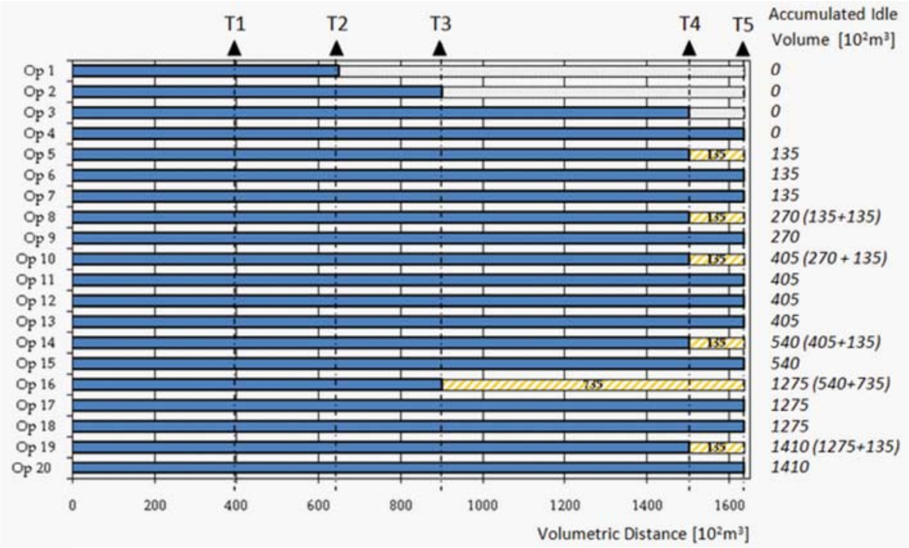


Table 2. Detailed delivery schedule for the first week using the NF-rule

Batch Injected	Batch Delivered	Terminal	Size [10²m³]	Start time [h]	End Time [h]
B6	B4	T2	90	5.00	15.00
	B4	T3	60	15.00	21.67
	B2	T4	80	21.67	30.56
	B1	T5	120	30.56	43.89
	B3	T4	10	43.89	45.00
	B1	T5	15	45.00	46.67
	B2	T5	50	46.67	52.22
B7	B3	T4	120	55.00	65.00
	B2	T5	5	65.00	65.42
	B4	T4	410	65.42	99.58
	B2	T5	65	99.58	105.00
	B3	T5	70	105.00	110.83
	B4	T5	5	110.83	111.17
	B5	T4	152	111.17	123.83
	B4	T5	73	123.83	130.00
	B7	T3	136	130.00	141.33
	B4	T5	62	141.33	146.58
	B5	T5	113	146.58	155.92
	B6	T4	10	155.92	156.75
	B5	T5	135	156.75	168.00

Figure 9. Delivery Schedule using the FF-rule for the first week of the monthly horizon

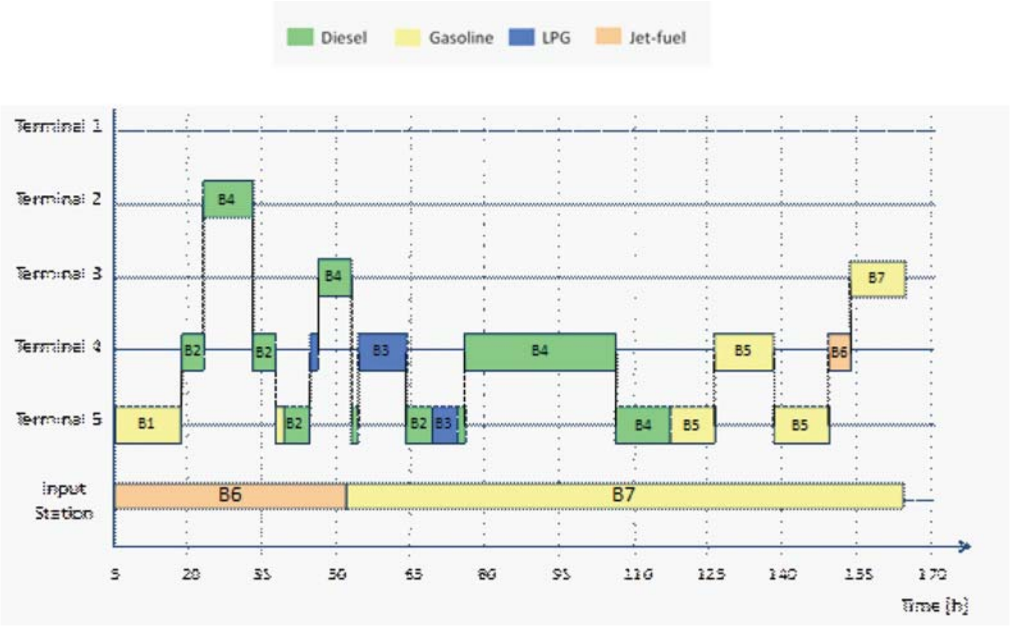


Figure 10. Active pipeline segments and evolution of the accumulated idle volume for the first week using the FF-rule

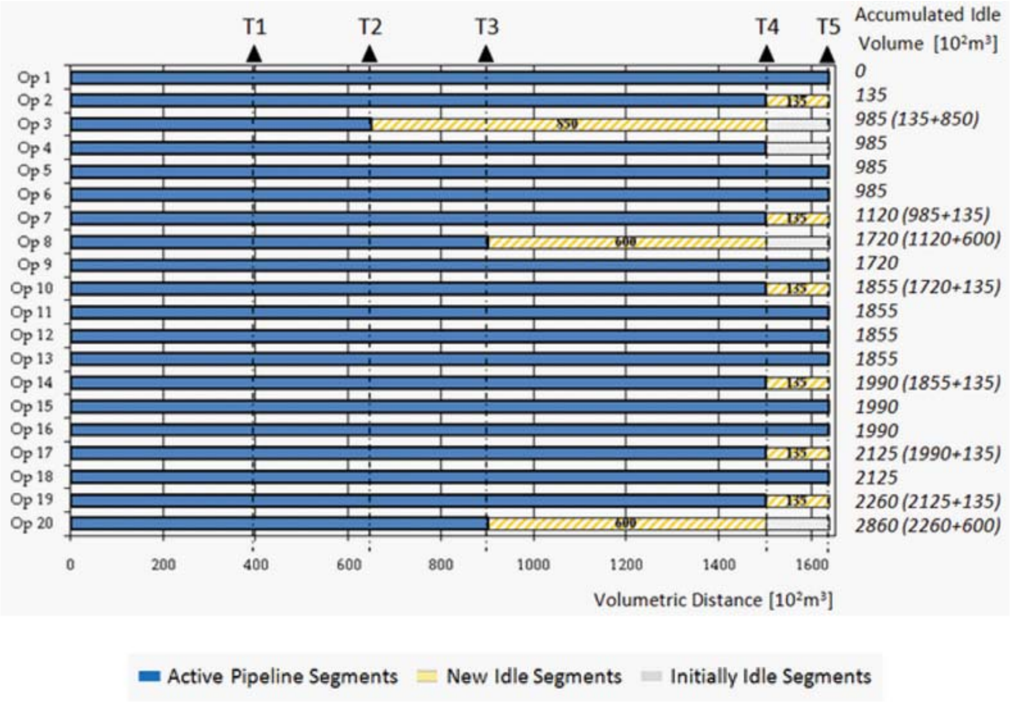


Table 3. Detailed delivery schedule for the first week using the FF-rule

Batch Injected	Batch Delivered	Terminal	Size [$10^2 m^3$]	Start Time [h]	End Time [h]
B6	B1	T5	120	5.00	18.33
	B2	T4	40	18.33	22.78
	B4	T2	90	22.78	32.78
	B2	T4	40	32.78	37.22
	B1	T5	15	37.22	39.00
	B2	T5	50	39.00	44.44
	B3	T4	10	44.44	45.56
	B4	T3	60	45.56	52.22
B7	B2	T5	5	55.00	55.42
	B3	T4	120	55.42	65.42
	B2	T5	65	65.42	70.92
	B3	T5	70	70.92	76.75
	B4	T5	5	76.75	77.08
	B4	T4	410	77.08	111.25
	B4	T5	135	111.25	122.58
	B5	T5	113	122.58	131.92
	B5	T4	152	131.92	144.58
	B5	T5	135	144.58	155.83
	B6	T4	10	155.83	156.67
	B7	T3	136	156.67	168.00

Figure 11. Delivery Schedule using the NF-rule for the monthly horizon

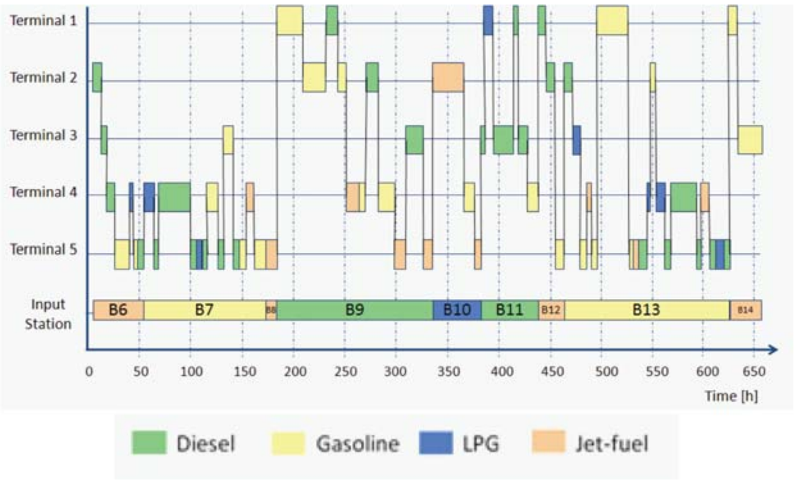
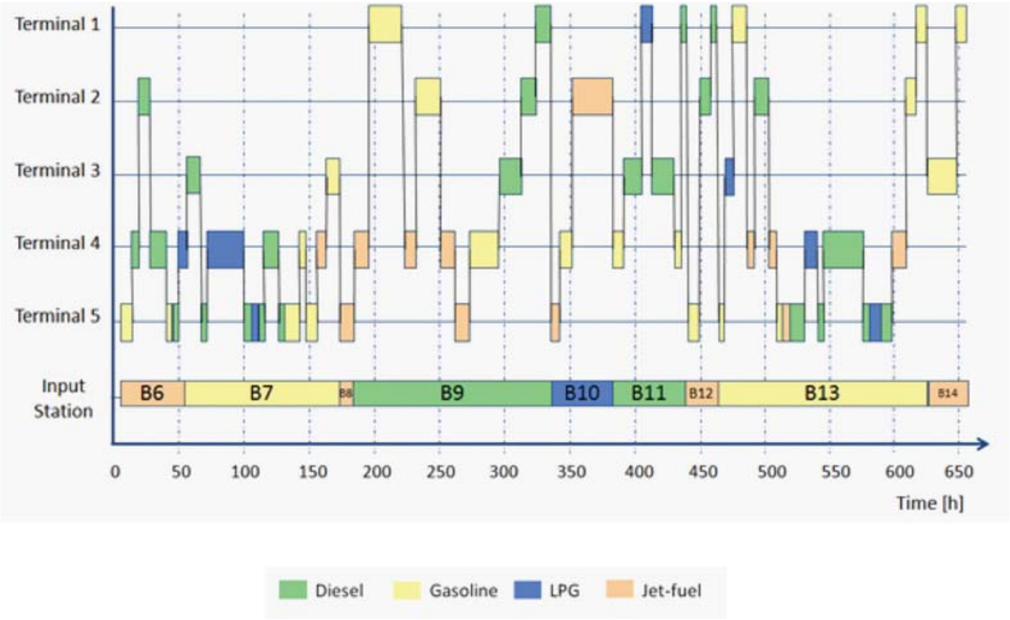


Figure 12. Delivery Schedule using the FF-rule for the monthly horizon



accumulated idle volume reaches up to 286,000 m^3, considerably higher than the 141,000 m^3 obtained with the *NF* rule. This difference is reduced over the following weeks of the monthly horizon. It can be observed from Figure 9 that the delivery of diesel product from *B2* to *T4* during the injection of *B6* is made through two non-consecutive runs. This situation arises because *T2* becomes a restricted terminal after injecting 160 units of batch *B6*.

Monthly Delivery Schedule

The complete monthly delivery schedules obtained with the *NF* and the *FF* rules are illustrated in Figures 11 and 12, respectively. The bottom line in both figures shows the pumping runs introducing batches *B6* to *B14* at the input station. The other lines depict the receiving tasks at every terminal while executing the corresponding batch injections. Both delivery schedules are then assessed according to the number of operations required to fulfill the product requirements, and the accumulated volume of idle product throughout the planning horizon. Using the *NF* priority rule for the whole monthly horizon, the accumulated volume of idle product is 1,404,500 m^3, and 65 cut operations are performed. On the other hand, when the *FF* rule is applied, the total volume of idle product rises to 1,472,500 m^3, but a slightly lower number of cut operations are required: 63.

CONCLUSION

This work addresses the scheduling of one of the most critical components in the petroleum supply chain: refined products pipeline networks. An advanced discrete event simulation model of a refined product trunk pipeline has been developed. The novel approach is very useful for validating operational pipeline schedules provided by rigorous optimization techniques. At the same time, it allows to generate and test alternative monthly product delivery schedules of single-source multiple-destinations pipeline networks in less than one minute of CPU time. Results show that different priority rules lead to significantly different delivery schedules, strongly affecting the cost-efficiency performance of pipeline operations. For instance, prioritizing nearest-to-refinery terminals may lead to a reduction in the total volume of idle pipeline segments, but the number of stripping operations (turning on/off pump stations) will increase. On the other hand, prioritizing the farthest terminals may cause an important growth in the number and volume of pipeline stoppages. It is quite clear that these very simple rules will rarely lead to the "optimal" delivery schedule. Nonetheless, the proposed logic structure permits to generate detailed delivery schedules by using alternative heuristic rules. In addition, it allows the visualization of the dynamic evolution of the pipeline system over time, using a friendly animated interface. The proposed approach can be easily extended to permit the use of simulation-based optimization tools in order to improve pipeline operations performance. Future work will be focused on developing efficient priority rules combined with heuristic search and rigorous formulations to find out cost-efficient and robust schedules of multiproduct pipeline networks with different configurations.

ACKNOWLEDGMENT

Financial support received from FONCYT-ANPCyT under Grant PICT 2006-01837, from CONICET under Grant PIP-2221, from Universidad Nacional del Litoral, Argentina, under Grant PI-66-337, from IN3 under project HAROSA and from AECID under Grant PCI D/024726/09 are fully appreciated.

REFERENCES

Boschetto, S. N., Felizari, L. C., Yamamoto, L., Magatão, L., Stebel, S. L., & Neves, F. Jr (2008). An integrated framework for operational scheduling of a real-world pipeline network. *Computer Aided Chemical Engineering*, *25*, 259–264. doi:10.1016/S1570-7946(08)80048-X

Cafaro, D. C., & Cerdá, J. (2004). Optimal scheduling of mulitiproduc pipeline system using a non-discrete MILP formulation. *Computers & Chemical Engineering*, *28*, 2053–2068. doi:10.1016/j.compchemeng.2004.03.010

Cafaro, D. C., & Cerdá, J. (2008). Dynamic scheduling of multiproduct pipelines with multiple delivery due dates. *Computers & Chemical Engineering*, *32*, 728–753. doi:10.1016/j.compchemeng.2007.03.002

Cafaro, D. C., & Cerdá, J. (2009). Optimal scheduling of refined products pipelines with multiple sources. *Industrial & Engineering Chemistry Research*, *48*, 6675–6689. doi:10.1021/ie900015b

García-Sánchez, A., Arreche, L. M., & Ortega-Mier, M. (2008). Combining simulation and tabu search for oil-derivatives pipeline scheduling. *Studies in Computational Intelligence*, *128*, 301–325. doi:10.1007/978-3-540-78985-7_12

Hane, C. A., & Ratliff, H. D. (1995). Sequencing inputs to multi-commodity pipelines. *Annals of Operations Research*, *57*, 73–101. doi:10.1007/BF02099692

Herrán, A., De la Cruz, J. M., & De Andrés, B. (2010). A mathematical model for planning transportation of multiple petroleum products in a multi-pipeline system. *Computers & Chemical Engineering*, *34*, 401–413. doi:10.1016/j.compchemeng.2009.11.014

Kelton, W. D., Sadowski, R. P., & Sturrock, D. T. (2007). *Simulation with ARENA* (4th ed.). New York, NY: McGraw-Hill.

Magatão, L., Arruda, L. V. R., & Neves, F. Jr. (2004). Mixed integer programming approach for scheduling commodities in a pipeline. *Computers & Chemical Engineering*, *28*, 171–185. doi:10.1016/S0098-1354(03)00165-0

Mori, F. M., Lüders, R., Arruda, L. V. R., Yamamoto, L., Bonacin, M. V., & Polli, H. L. (2007). Simulating the operational scheduling of a real-world pipeline network. *Computer Aided Chemical Engineering*, *24*, 691–696. doi:10.1016/S1570-7946(07)80138-6

Moura, A. V., de Souza, C. C., Cire, A. A., & Lopez, T. M. (2008). Planning and scheduling the operation of a very large oil pipeline network. In P. J. Stuckey (Eds.), *Proceedings of the 14th International Conference on Principles and Practice of Constraint Programming* (LNCS 5202, pp. 36-51).

Neves, F. Jr, Magatão, L., Stebel, S. L., Boschetto, S. N., Felizari, L. C., & Czaikowski, D. I. (2007). An efficient approach to the operational scheduling of a real-world pipeline network. *Computer Aided Chemical Engineering*, *24*, 697–702. doi:10.1016/S1570-7946(07)80139-8

Rejowski, R., & Pinto, J. M. (2003). Scheduling of multiproduct pipeline system. *Computers & Chemical Engineering*, *27*, 1229–1246. doi:10.1016/S0098-1354(03)00049-8

Rejowski, R., & Pinto, J. M. (2008). A novel continuous time representation for the scheduling of pipeline systems with pumping yield rate constraints. *Computers & Chemical Engineering*, *32*, 1042–1066. doi:10.1016/j.compchemeng.2007.06.021

Sasikumar, M., Prakash, P. R., Patil, S. M., & Ramani, S. (1997). Pipes: A heuristic search model for pipeline schedule generation. *Knowledge-Based Systems*, *10*, 169–175. doi:10.1016/S0950-7051(97)00026-9

Zyngier, D., & Kelly, J. D. (2009). Multi-product inventory logistics modeling in the process industries. *Optimization and Logistics Challenges in the Enterprise*, *30*, 61–95. doi:10.1007/978-0-387-88617-6_2

M. Fernanda Gleizes is a former researcher in the department of industrial engineering at Universidad Nacional del Litoral (UNL). She received her BS in industrial engineering from Universidad Nacional del Litoral-Argentina in 2008. Her research contributions have been in the area of oil pipeline planning and scheduling. She is currently working as Process Engineer in the production area of Cervecería Santa Fe.

Germán Herrero is a former teaching assistant in the department of industrial engineering at Universidad Nacional del Litoral (UNL). He received his BS in industrial engineering from Universidad Nacional del Litoral-Argentina in 2008. His research contributions have been in the area of oil pipeline planning and scheduling. He is currently working as Process Engineer in the production area of Cervecería Santa Fe.

Diego Cafaro is professor of industrial engineering at Universidad Nacional del Litoral (UNL), and posdoctoral fellow at the Argentine National Scientific and Technical Research Council (CONICET). He received his PhD in chemical technology from UNL in 2008, and his B.S. in Industrial Engineering from Universidad Nacional del Litoral in 2003. His research contributions have been in the areas of oil pipeline planning and scheduling. His teaching interests include courses on Modeling and Optimization in Process Systems Engineering.

Carlos A. Méndez is professor of industrial engineering at Universidad Nacional del Litoral (UNL), and associate researcher at the Argentine National Scientific and Technical Research Council. He received his PhD in engineering from UNL in 2002, and his B.S. in Information Systems Engineering from Universidad Tecnológica Nacional-Argentina in 1998. He held Visiting Professor Positions at Carnegie Mellon University, Pittsburgh-EE.UU. (2004), Instituto Superior Técnico, Lisbon-Portugal (2005,2010), DECRC, ABB Corporate Research Center, Ladenburg-Germany (2005), Universidad de Valladolid, -Spain (2006–2008), Jozef Stefan Institute, Ljubljana, Slovenia (2006), Universitat Politècnica de Catalunya, Barcelona-Spain (2005, 2006, 2008–2009) Universitat Rovira i Virgili, Tarragona-Spain (2009) and Universidad de Cantabria, Santander-Spain (2009). His research contributions have been in the areas of planning and scheduling, optimal process synthesis, design and operation, supply chain operational management, oil pipeline planning and scheduling, dynamic vehicle routing and pickup & delivery problems. His teaching interests include courses on Modeling, Simulation and Optimization in Process Systems Engineering.

Jaime Cerdá is professor of chemical and industrial engineering at Universidad Nacional del Litoral (UNL), and superior researcher at the Argentine National Scientific and Technical Research Council (CONICET). He received his PhD in chemical engineering from Carnegie-Mellon University in 1980, his MS in chemical engineering from Carnegie-Mellon University in 1979, and his BS in chemical engineering from Universidad Nacional del Litoral at Argentina in 1968. He is a former Industrial Engineering Department Head of the School of Chemical Engineering at UNL. He held Visiting Professor positions at Universidad de Valladolid, Spain (2006), Carnegie-Mellon University (1982, 1995), Universidad Católica del Norte at Antofagasta, Chile (1995) and Universidad de Barcelona at Spain (1987). His research contributions have been in the areas of planning and scheduling of batch and continuous processes, energy integration, process synthesis, supply chain operational management, oil pipeline planning and scheduling, dynamic vehicle routing and pickup & delivery problems.

International Journal of Information Systems and Supply Chain Management

An official publication of the Information Resources Management Association

Mission

The *International Journal of Information Systems and Supply Chain Management* (IJISSCM) provides a practical and comprehensive forum for exchanging novel research ideas or down-to-earth practices which bridge the latest information technology and supply chain management. IJISSCM encourages submissions on how various information systems improve supply chain management, as well as how the advancement of supply chain management tools affects the information systems growth. The aim of this journal is to bring together the expertise of people who have worked with supply chain management across the world for people in the field of information systems.

Subscription Information

IJISSCM is published quarterly: January-March; April-June; July-September; October-December by IGI Global. Full subscription information may be found at www.igi-global.com/IJISSCM. The journal is available in print and electronic formats.

Copyright

Correspondence and questions:

Editorial:
John Wang, PhD
Editor-in-Chief
IJISSCM
j.john.wang@gmail.com;
journalswang@gmail.com

Subscriber Info:
IGI Global
Customer Service
701 E Chocolate Avenue
Hershey PA 17033-1240, USA
Tel: 717/533-8845 x100
E-mail: cust@igi-global.com

IJISSCM is currently listed or indexed in: Bacon's Media Directory; Cabell's Directory; CSA Illumina; DBLP; GetCited; Google Scholar; The Index of Information Systems Journal; INSPEC; Library and Information Science Abstracts (LISA); MediaFinder; Norwegian Social Science Data Services (NSD); The Standard Periodical Directory; Ulrich's Periodicals Directory