NONLINEAR MODELS FOR CRUDE OIL SCHEDULING ON PORTS

Fabio Dias Fagundez[a], João Lauro Dorneles Facó[b], Adilson Elias Xavier[c]

[a] Advanced Optimization Laboratory - Federal University of Rio de Janeiro (UFRJ)
[b] Department of Computer Science - Federal University of Rio de Janeiro (UFRJ)
[c] Systems Engineering and Computing Program - Federal University of Rio de Janeiro (UFRJ)

[a] fabio.fagundez@ufrj.br, [b] jldfaco@ufrj.br , [c] adilson@cos.ufrj.br

ABSTRACT
In this paper we present a continuous nonlinear programming approach to crude oil scheduling on ports. The main idea is to consider the schedule as a dynamic system which must operate under certain constraints. As the optimal control theory is well-fit for optimizing the operation of dynamic systems, our scheduling problem is modeled as an optimal control problem, where transfer operations are carried out by flows from the ships to the port facilities (tanks). Such flows are mapped as control variables, whilst equipment contents (volume and properties) are mapped as state variables. Yes-No decisions are modeled as complementarity constraints, instead of with the use of discrete variables, allowing a continuous nonlinear model. We illustrate this approach with computational examples, which were solved to local optimality in reasonable computational time.

Keywords: port scheduling, crude oil scheduling, mathematical programming

1. INTRODUCTION

The maritime transportation is responsible for an almost-monopoly on the Import & Export international good transactions, with an increasing importance year after year. During the last decade, this industry has been pressured to adopt modern management techniques, in order to succeed in face of new challenges posed by global commerce, market deregulamentation, and companies' merges (Christiansen et al. 2004). On one hand, shippers must use the full capacity of their fleets, as they cannot afford having empty vessels laying still in ports, waiting for new customers. On the other hand, customers who rent or hire ships must know that the vessels will be available in time to attend their contracts. This complex fleet management is a rich field for optimization techniques. One of the bottlenecks of maritime transportation is the handling of ships in ports, considering the berthing and cargo loading/unloading processes: a vessel must be handled within a given time-frame, in order to minimize demurrage fees. There is a limited number of berth places and the port storage must be ready to handle the ship without incurring in delays. For instance, in 2006, the maritime transportation costs in Brazil ammounted to US$ 7 billions, from which US$ 1.5 billions were spent with demurrage fees only, caused by delays in ports (Collyer 2006).

This work addresses the schedule of ships in ports, considering the storage facilities, jetties and operational constraints, such as jetty unavailability and load/unloading limitations. A widespread approach for scheduling problems is the use of mixed-integer programming models, which formulate discrete decisions as 0-1 variables. The main advantage of this approach is that, in the case of mixed-integer linear programming (MILP) problems, when an optimal is found, there is a guarantee that it is a global optimum – as in the Linear Programing case. The main disadvantage of such models is the computational burden on real-world instances, which usually end in huge MIP models that may not be solved to optimality in reasonable time. Bearing that in mind, we decided to model the problem from a different angle, noticing that a scheduling system is actually a dynamic system, susceptible to an optimal (controlled) operation. Control variables are the loading/unloading flows, and state variables are the inventory on ships and port facilities. Moreover, schedule decisions can be modeled as complementarity constraints on the control variables. The main advantage of this formulation is that the problem does not suffer from an exponential growth with the addition of time intervals, whereas the main disadvantage is that the problem, although continuous, becomes nonconvex (with many local minima).

We tested the proposed approach with crude oil scheduling in ports problems (Figure 1), and compared to an alternative MILP formulation, based on the literature.

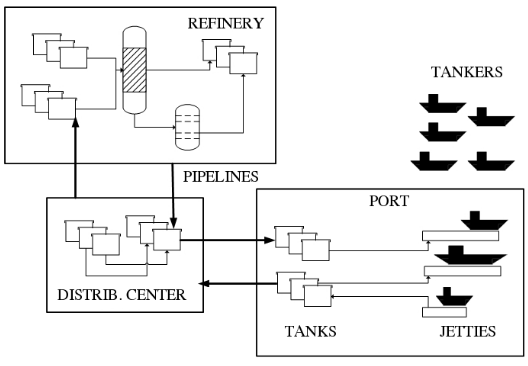

Figure 1: Crude Oil Scheduling

2. CRUDE OIL SCHEDULING ON PORTS

Planning and scheduling are activities of major economic importance. Efficient planning and scheduling can optimize resources usage, reduce waste and increase operational profits. They can avoid delays and assure that demands are supplied without degrading the quality of services. It is well-known that to find an optimal schedule is a NP-Complete problem (Más and Pinto, 2003). Therefore, many companies build some feasible schedules only and then make comparisons among them. The use of computational techniques such as simulation and mathematical optimization can help the decision-maker, as they can fasten the evaluation of different scenarios and find optimal or sub-optimal schedules. In this paper we propose a dynamic optimal control model that can be used in schedule simulations as well as in scheduling optimization.

Amid various planning and scheduling problems, we focus on the scheduling of crude oil on ports. This problem is of major importance for oil companies, as it directly affects the refineries' production plan. One can divide the logistic system related to crude oil and derivatives production and distribution in three subsystems: harbor, distribution center (intermediate storage), and plants or refineries (Figure 1). Each subsystem is connected to the others by pipelines, and the sequence of transferred material in the pipelines is how a subsystem affects the scheduling of the others.

The plant or refinery subsystem is very complex, while distribution centers coordinate the distribution of crude oil among different refineries. One example is the problem studied in (Más and Pinto, 2003), where the center acts as a buffer between the maritime terminals and four refineries. Depending on the desired scope, it is possible to consider each subsystem separately as subject to optimization, and to represent the others as external inputs to the problem.

Our objective is to schedule the ships within the port, and the subsequent port tank operations to unload their cargoes. Therefore, we detail the port subsystem only, considering planned crude oil volumes on the pipelines – to meet the refineries' demand – and known estimated time of arrival (ETA) for all ships. In many companies this is actually the case (Más and Pinto, 2003).

The port has its own park of tanks, large enough to store material pumped to or from the vessels for a certain number of days. The tanks may belong to the same company of the materials or may be hired from third parties. Even if the renting costs are negligible, the cost of immobilized capital is not. Therefore, stocks represent losses of interests on the value of the stored material, and stocking costs have to be considered in the problem formulation.

Tanks are dedicated to certain kind of materials, which can be crude oil or finished products. In the case of finished products, it is common to inspect the product's quality inside the tank, and to seal the tank to assure that no new product is going to be blended and modify its quality. In the case of crude oils, it is common to find impurities in the oil, such as salt, water and minerals. Therefore, tanks must let the oil (which they had received) to settle during a certain period of time in order to segregate the impurities. From these two examples, an operational rule is inferred: no tank in the harbor should be able to start a new delivering transfer if it had not been idle for a certain amount of time. This time is usually called "settling time" or simply "idle time". Another rule to operate tanks is not to use two tanks in parallel, transferring to a common final destination as it would cause blending inside the pipes.

The port is composed by a certain number of jetties, each one with defined draught and extension, and pumps that may be restrictive on what material to transfer. Therefore, ships can berth only on a restricted set of jetties, defined by their cargo or physical dimensions. The period of time that a vessel occupies a jetty is calculated as the following: time to approach the berthing place, plus the pumping time to the tanks, plus the time to leave the jetty. Therefore, the jetty will be available for another ship only a few hours after the previous ship stopped pumping. If the vessel cannot be completely processed until its contractual leaving date, the contractor is penalized (demurrage). One of the main goals of this scheduling problem is to minimize the demurrage.

3. THE MODEL

The fundamental scheduling activity is the transfer operation, defined by a single source equipment and a single destination equipment, and a flow of material from the source to the destination. The control vector $u(t_i)$ is the vector whose each entry stands for a possible flow rate between two equipments at a given time ti, subject to upper and lower bounds and operational constraints. Basically, one can consider a schedule infrastructure as a graph (Figure 2), whose arcs are possible flows, and whose nodes are equipments. Therefore, every positive entry $u_j(t_i)$ stands for the scheduled flow rate of a transfer operation at time t_i, in arc j, which connects a pair of equipments. The optimization problem is to define a feasible sequence of $u(t_i)$, for all time instants t_i, which minimizes an objective function J. Operational constraints, such as "One equipment cannot be the destination of two transfer operations at the same time, in order to avoid blending in the inlet lines" can be modeled in different manners.

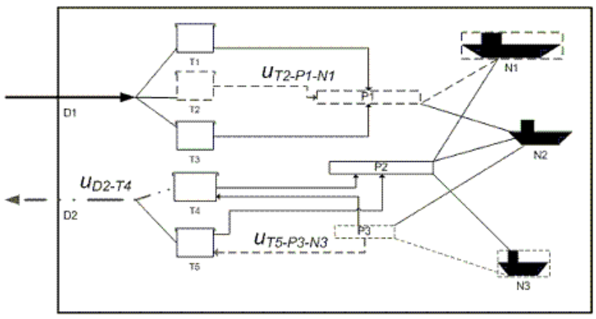

Figure 2: Problem structure as a graph

Table 1 presents two possible approaches: (a) MILP, and (b) complementarity based (NLP). A_N is the set of indexes for all arcs whose destination is equipment N. As N can participate in at most one transfer operation at time t_i, either all flows in the A_N arcs are zero at t_i (no transfer happens with destination N at time t_i), or only one flow is greater than zero at t_i (N is the destination of only one transfer operation at time t_i).

Table 1: Modeling Approaches

Type	Model
MILP	$\sum_j b_j(t_i) \leq 1$ $0 = b_j(t_i)\, u_j^{min} \leq u_j(t_i) \leq b_j(t_i)\, u_j^{max}(t_i)$ $u_j(t_i) \in R^n,\ b_j(t_i) \in \{0, 1\},\ j \in A_N$
NLP	$\sum_{k \in AN} \sum_{j > k \in AN} u_k(t_i)\, u_j(t_i) = 0$ $0 = u_j^{min} \leq u_j(t_i) \leq u_j^{max}(t_i)$ $u_j(t_i) \in R^n,\ j \in A_N$

Model (a) is a typical mixed-integer program (Shah 1996, Más and Pinto 2003), which requires an additional control vector $b(t_i)$ of binary variables, whose each entry is directly related to the $u(t_i)$ entry of same index. If $b_j(t_i)$ is set to 1 (one), then a flow is allowed through arc j; otherwise, no flow is allowed through arc j. This is enforced by the manipulation of the upper and lower bounds on $u_j(t_i)$: if $b_j(t_i) = 1$, the bounds are preserved; otherwise, they are set to zero. The summation constraint guarantees that at most one $b_j(t_i)$ can be evaluated as 1 (one) at t_i, $j \in A_N$. All others binary variables associated to A_N must be evaluated as zero.

Model (b) is a complementarity-based nonlinear program, which relies on the control vector $u_j(t_i)$ only. There is no need for additional binary variables. The summation of the products of all A_N flows two by two is equal to zero if and only if all flows are equal to zero or only one flow is greater than zero, making N as the destination of at most one transfer operation, as required. The main disadvantage of this formulation is that it defines a nonconvex nonlinear program.

A point defined by the MILP formulation can be transformed in an NLP point by simply removing the binary variables from the problem and replacing the mixed-integer constraints for the complementarity ones. A point defined by the NLP formulation can be transformed in an MILP point by simply adding the binary variables and replacing the complementarity constraints by the mixed-integer ones. For each positive flow, the corresponding binary variable is set to 1 (one), while for each naught flow, the corresponding binary variable is set to 0 (zero). A feasible point in the NLP is equivalent to a feasible in the MILP. The opposite is not necessarily true: if the NLP features additional nonlinear constraints that could not be equivalently modeled in the MILP, a feasible MILP point may not be a feasible NLP point.

Based on these foundations, one can analogously extend the problem to consider many other constraints and variables. In order to illustrate this, we model the crude oil scheduling problem with both approaches. One wants to determine: (i) tanker allocation on the port jetties; (ii) transfer operations between ships, tanks, and pipelines; (iii) sequence of pipeline parcels (end products and crude oil), in such a manner that an objective cost function is minimized and operational constraints are respected. The problem considers: (i) crude oil demand as known; (ii) ETA are given for all tankers. Ideally, a good schedule would use a small number of tanks, but it is important to notice that inventory costs are secondary when compared to the cost of not meeting a refinery production plan. Jetties can be restrictive on what vessels and cargoes they handle, in accordance to their dimensions (draught and length) or pumping capacity. A ship must berth, unload (or load), and leave the port during the time window defined by contract and she can berth in a jetty only if the ship that had previously used this jetty had enough time to leave the port. An ideal schedule must minimize the demurrage costs. Other constraints are considered as well: (i) blending operations are not allowed in lines, i.e., each transfer operation has only one source equipment and one destination equipment; (ii) a tank can deliver material to another equipment only if the necessary "settling time" or "idle time" has been observed (e.g. to separate brine from crude oil or to assure the lab analysis of an end product).

The NLP model is composed by control variables (Eq. 2), state variables (Eq. 3), state equations (Eq. 4, 5, and 6), and complementarity equations to model scheduling decisions: unique definition of source and destination in a transfer operation (Eq. 7), idle time to segregate impurities (Eq. 8), berthing time (Eq. 9), and constant flow constraints (Eq. 10). Both control and state variables are bounded by upper and lower limits. The objective-function (Eq. 1) is a summation of different costs, that can be prioritized with the use of weights. In the case of crude oil scheduling, we considered the following costs: demurrage (Eq. 11), unattained demand (Eq. 12), inventory (Eq. 13), and changeovers (Eq. 14).

$$\min J = \sum_{cost} w_{cost} C_{cost} \quad (1)$$

$$u_{min} \leq u(t_i) \leq u_{max}(t_i) \quad (2)$$

$$x_{min} \leq x(t_i) = [v(t_i)\ p(t_i)]^T \leq x_{max} \quad (3)$$

$$v(t_i) = v(t_{i-1}) + (t_i - t_{i-1})Uu(t_{i-1}) \quad (4)$$

$$p(t_i) = F(x(t_{i-1}), u(t_{i-1}), t_i - t_{i-1}) \quad (5)$$

$$p^{density}_n(t_i) = (u_{inlet,n}(t_{i-1})\, p^{density}_{inlet,n}(t_{i-1})(t_i - t_{i-1}) + v_n(t_{i-1})\, p^{density}_n(t_{i-1}))/(u_{inlet,n}(t_{i-1})(t_i - t_{i-1}) + v_n(t_{i-1})) \quad (6)$$

$$0 = r_n(t_i) = \sum_{k \text{ in } Jn} \sum_{j > k \text{ in } Jn} u_k(t_i) u_j(t_{i-1}) \quad (7)$$

$$0 = z_n(t_i) = \sum^{ti-1}_{t' = ti-Tidle} \sum_{k \text{ in } Jout,n} \sum_{j \text{ in } Jin,n} u_k(t_{i-1}) u_j(t') \quad (8)$$

$$0 = s_n(t_i) = \sum^{ti-1}_{t' = ti-Tberth} \sum_{j \text{ in } Jn,p} \sum_{j \text{ in } Jn,!p} u_k(t_{i-1})u_j(t') \quad (9)$$

$$0 = q_n(t_i) = u_{0,n} - \sum_{j \text{ in } Jn} u_j(t_{i-1}) \quad (10)$$

$$C_{demurrage} = \sum_{j \text{ in Ships}} \sum_t c_j^{demur} \text{ cargo\_gap}_j(t) \quad (11)$$

$$C_{demand} = \sum_{j \text{ in Pipelines}} \sum_{p \text{ in Products}} c_j^{demand}(\text{demand}_{j,p} - \sum_t^T v_{j,p}(t)) \quad (12)$$

$$C_{inventory} = \sum_{j \text{ in Storages}} \sum_t c_j^{inventory} v_j(t) \quad (13)$$

$$C_{changeover} = \sum_j \sum_t c_j^{changeover}(u_j(t_i) - u_j(t_{i-1}))^2 \quad (14)$$

Note: Eq. 4 features U as an incedence square matrix with entries in {0, 1, -1}. Eq. 5 is a general blending functional, while Eq. 6 is a particular blending function (for density), which is the one we employ in our test cases. Eq. 7 enforces that only one flow can be used by an equipment n at time t_i, therefore, a transfer operation has only one source and only one destination at time t_i. Eq. 8 enforces the idle time, while Eq. 9 enforces the necessary berthing time for ships. Eq. 10 force a constant flow $u_{0,n}$ for a given equipment n – this constraint can be easily changed to force a variable flow, if needed. Eq. 11 deals with demurrage cost: we do not employ the classic demurrage formulation, but one that is also proportional to the remaining volume to be transferred that is delayed (cargo_gap). Eq. 14 accounts for changeovers in the problem, by penalizing the differences in the flows. In all cost equations cjcost is a different arbitrary unitary cost.

The MILP model is similar to the NLP model, with the replacement of Equations 2, 7, 8, and 9 by Equations 2a, 7a, 8a, and 9a. Complementarity constraints were replaced by mixed-integer constraints, featuring binary decision variables (b_j), leading to larger models. This model was based in the crude oil scheduling literature, mainly (Shah 1996) and (Más and Pinto 2003):

$$D(b_j(t_i))u_{min} \leq u(t_i) \leq D(b_j(t_i))u_{max}(t_i), \;\; b_j(t_i) \text{ in } \{0,1\} \quad (2a)$$

$$r_n(t_i) = \sum_{j \text{ in } Jn} b_j(t_{i-1}) \leq 1 \quad (7a)$$

$$z_n(t_i) = \sum_{j \text{ in } Jout,n} b_j(t_{i-1}) + \sum^{ti-1}_{t' = ti-Tidle} \sum_{j \text{ in } Jin,n} b_j(t') \leq 1 \quad (8a)$$

$$s_n(t_i) = \sum_{j \text{ in } Jn,p} b_j(t_{i-1}) + \sum^{ti-1}_{t' = ti-Tberth} \sum_{j \text{ in } Jn,!p} b_j(t') \leq 1 \quad (9a)$$

Note: Eq. 4a features a diagonal matrix $D(b_j(t_i))$, composed by the binary variables b_j, which are added to the model in the MILP formulation. These variables represent scheduling decisions: there is no flow u_j at time t_i if $b_j = 0$ at time t_i, and there is a flow u_j if $b_j = 1$. Equations 7a, 8a, and 9a model represent the following constraints: only one flow can be used by an equipment n at time t_i, idle time, and berthing time constraints.

4. RESULTS

Typical NLP methods can solve problems with complementarity constraints efficiently (Leyffer 2005). As the formulation is nonconvex, the method initialization is an important issue. We define trivial points as the points where the control vector *u* is equal to zero at all time instants. These points are very easy to be constructed, but not feasible in the original problem formulation. At trivial points, the demurrage costs are maximum and the norms of the additional states are $||z|| = ||r|| = ||s|| = 0$ and $||q|| >> 0$. However, a trivial point is feasible in a relaxation of the original model, if we remove equations (7) to (10) from the constraints, and add them as penalties on the objective function, obtaining a merit function:

$$J' = J + M \sum_t e^T r(t) + e^T s(t) + e^T z(t) + e^T q(t) \quad (15)$$

A penalty method based on the optimization of successive relaxed problems with merit function J' can solve the original problem, converging to local minimum. A multistart procedure can be employed as well in the initialization of the relaxed problems.

Five test instances were coded in AMPL (Fourer et al. 2003). The MILP and NLP instances share a common objective function, and do not feature equations (5) and (6) as a means to allow the comparison between their optimal solutions. It is clear that the NLP model is more compact than the MILP model, as shown in Table 2. The number of variables is the one determined after AMPL's pre-solve procedure, which eliminates redundancies.

Table 2. Models' dimensions

	MILP		NLP	
	Binary/ Continuous variables	Linear Cons-traints	Linear / Nonlinear variables	Linear / Nonlinear constraints
1(A)	12 / 25	31	12 / 19	19 / 6
1(B)	12 / 25	36	12 / 19	24 / 6
2	34 / 82	111	34 / 77	64 / 23
3(A)	93 / 158	265	93 / 76	92 / 11
3(B)	93 / 158	275	57 / 78	86 / 11

No specialized software was employed to optimize the test instances, but only standard commercial solvers:

- For MILP cases: CPLEX (v. 10.1.0) (ILOG 2002);
- For NLP cases: SNOPT (v. 6.1) (Gill et al. 2002), and MINOS (v. 5.5) (Murtaugh e Saunders 1982).

All cases were initialized at trivial points – as previously defined – and solved on a workstation with the following configuration: Intel Core Duo T2250 1.73GHz, RAM 1GB, Linux OpenSUSE 10.1. All cases were solved in reasonable time to optimality: around 1 second. Moreover, the solutions found in all cases were able to handle the ships with no delays – as desired – and meet the refineries crude oil demand – as desired as well.

The test cases are detailed as follows:

- **Test 1:** Two crude tanks and one pipeline connected to a refinery, whose crude demand has to be fulfilled. There are 2 configurations for this scenario: (A) allows the pipeline to be idle in certain periods, (B) keeps the pipeline with a constant flow, during the entire schedule. The MINOS run converged to a local minimum in (B) configuration. The NLP objective function is (Eq. 15), while the MILP is (Eq. 1). The penalty parameter M was set as 1.0.

- **Test 2:** Two crude tanks, one jetty and two tankers, whose cargo had to be unloaded. The NLP objective function is (Eq.15) with M=1.0, while the MILP is (Eq.1).
- **Test 3:** Three crude tanks, one jetty, three tankers, whose cargo had to be unloaded, and one pipeline, whose demand has to be fulfilled. There were 2 configurations: (A) allows the pipeline to be idle in certain periods, (B) keeps the pipeline with a constant flow. The SNOPT run converged to a local minimum with demurrage costs in (B) configuration. The NLP objective function is (Eq. 15) with M=0.05, while the MILP is (Eq. 1).

Table 3 shows that the optimal solutions were achieved in both models (MILP and NLP).

Table 3. MILP and NLP results

	MILP	NLP	
	CPLEX	MINOS	SNOPT
1(A)	J = 1460 14 iterations	J = 1460 17 iterations	J = 1460 51 iterations
1(B)	J = 1600 13 iterations	J = 1625 5 iterations	J = 1600 13 iterations
2	J = 0.33 63 iterations	J = 0.33 191 iterations	J = 0.33 12 iterations
3(A)	J = 0 324 iterations 8 B&B nodes	J = 0 411 iterations	J = 0 812 iterations
3(B)	J = 0 397 iterations 25 B&B nodes	J = 0 472 iterations	J = 18.27 544 iterations

The number of iterations are similar for both models, but, as larger problems are approached, the MILP branch and bound procedure expands a larger number of nodes. As the complementarity model is nonconvex, a nonlinear programming method, such as MINOS and SNOPT, may converge to local optima, differently from what happens with the mixed-integer model when solved by a typical branch-and-bound method, such as CPLEX. On the other hand, the complementarity model is more compact, featuring less variables and constraints than the MILP one.

Noticing that one NLP solution is equivalent to an MILP feasible point, we propose a hybrid scheme (Figure 3): solve the continuous NLP problem and then transform its solution as an initial point for the MILP. If needed, call NLP runs in difficult nodes of the MILP B&B tree. This scheme may be able to reduce the total number of branches and Simplex iterations in the MILP optimization, as the NLP point is an integral MILP good solution.

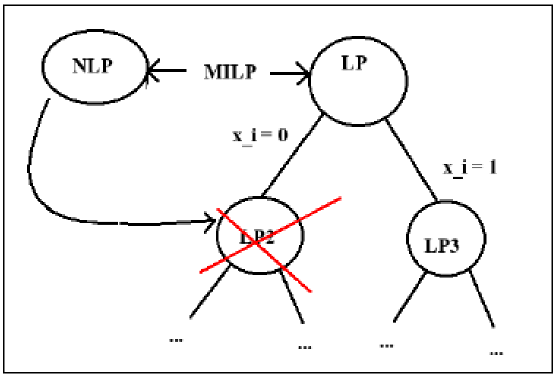

Figure 3: Hybrid scheme

At the current state of our research, we employed the NLP solutions to initialize the MILP models, and compared the number of iterations and branched nodes. A substantial reduction in the number of iterations in the MILP optimization run is detected (Table 4).

Table 4. MILP cases with trivial and NLP starts

	MILP Iterations		
	$(x,u)^0=(x^0, 0)$	$(x,u)^0=(x,u)^{SNOPT}$	$(x,u)^0=(x,u)^{MINOS}$
1 (A)	14	13	13
1 (B)	13	4	4
2	63	47	40
3 (A)	324 (8 nodes)	215	215
3 (B)	397 (25 nodes)	265 (6 nodes)	215

5. CONCLUSIONS

One of the bottlenecks in maritime transportation industry is how ships are handled in ports: delays are very common and expensive, incurring in high demurrage costs. We proposed a NLP model for this problem, and presented some computational examples in order to illustrate the suitability of the proposed model. Oil companies, port administrators, and freighter companies are possible users of this model. The most important advantage of using mathematical models for this problems is to find schedules that incurr in no demurrage costs and meet all operational constraints. All cases were solved in reasonable time to optimality (around 1 second), showing that this approach is an interesting option for industry applications. Finally, it is important to notice that the complementarity equations herein described can be employed in others scheduling problems.

REFERENCES

Christiansen, M., Fagerholt, K., Ronen, D., 2004. Ship routing and scheduling: status and perspectives *Transportation Science*, 38 (1), 1-18.

Collyer, W. , 2006. Sobreestadia de navios: a regra 'once on demurrage, always on demurrage' *Jus Navigandi*, 10 (1166) 10. [in Portuguese]

Fourer, R. Gay, D.M. and Kernighan, B.W., 2003. *AMPL: A Modeling Language for Mathematical Programming.* Pacific Grove: Duxbury Press Brooks/Cole-Thomson Publishing Company.

Gill, P. E., Murray,W. and Saunders, M.A., 2002 SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, 12 (4) 979.

ILOG, Inc., 2002. *ILOG AMPL/CPLEX System Version 8.0 User's Guide*. ILOG, Inc.

Leyffer, S. (2005) Complementarity Constraints as Nonlinear Equations: Theory and Numerical Experience. *Technical Report ANL/MCS-P1054-0603*, Argonne National Laboratory, Argonne.

Más, R., Pinto, J.M., 2003. A mixed-integer optimization strategy for oil supply in distribution complexes. *Optimization and Engineering*, 4 (1) 23.

Murtagh B.A. and Saunders M.A., 1982. A projected Lagrangian algorithm and its implementation for sparse non-linear constraints. *Mathematical Programming Studies*, 16, 84.

Shah N., 1996. Mathematical Programming Techniques for Crude Oil Scheduling. *Computers and Chemical Engineering*, 20 (sup.), S1227-S1232.