

Continuous Nonlinear Programming Techniques to Solve Scheduling Problems

Fabio D. FAGUNDEZ, Adilson E. XAVIER, Joao L.D. FACO

Universidade Federal do Rio de Janeiro
Av. Horácio de Macedo, 2030
Centro de Tecnologia, PESC/COPPE, Bloco H, sala H-318
Rio de Janeiro, RJ, Brazil. 2941-914
e-mail: fabio.fagundez@ufrj.br, fagun@bndes.gov.br

Received: July 2008; accepted: February 2009

Abstract. One of the most known applications of Discrete Optimization is on scheduling. In contrast, one of the most known applications of Continuous Nonlinear Optimization is on the control of dynamic systems. In this paper, we combine both views, solving scheduling problems as dynamic systems, modeled as discrete-time nonlinear optimal control problems with state and control continuous variables subjected to upper and lower bounds. Complementarity constraints are used to represent scheduling decisions. One example we discuss in detail is the crude oil scheduling in ports, with numerical results presented.

Keywords: nonlinear programming, optimal control, scheduling.

1. Introduction

Scheduling problems can be modeled as discrete optimization problems, as they feature two general types of constraints: discrete constraints and continuous constraints. The first group relates to enumerative or logical decisions like “choose source A to send cargo B to destination C at time t ”, whereas the second relates to more general limitations like “the maximum storage capacity of store A is $30,000 \text{ m}^3$ ”. Constraints on discrete variables stand for assignment and sequencing decisions, and continuous equations model mass, volume, energy or component balances.

Floudas and Lin (2005) recently presented a survey on process scheduling, where they emphasize the importance of Mixed Integer Linear Programming (MILP) in this field. The guarantee of global optimality is considered as the highlight of this approach. However, due to scheduling’s NP-completeness, such models suffer from the curse of dimensionality (the number of variables is exponential to number of time periods), and MILP solving procedures reach unacceptable computational times to find a solution for a real-world problem. A possible approach to reduce computational time is to develop tailor-made heuristics that can find good solutions for specific scheduling problems, based on the problems’ particular structures as in Al-Yakoob and Sherali (2007) and Janiak and Kovalyov (2006). Another possible approach is to substitute typical scheduling objectives to

alternative objectives to be optimized, which may feature better mathematical properties: one example is found in the work of Dzemyda (1999). Additionally, the research community has been constantly working on formulations to reduce models' dimensions, particularly within nonuniform time-discretization frameworks (see Floudas and Lin (2004) for a thorough discussion on this subject). Moreover, nonlinear phenomena are dealt with linear approximations, relaxed or removed from the models. In this paper a nonlinear programming (NLP) formulation based on continuous variables is proposed, trying to achieve reasonably small models and to converge to local optimal solutions in affordable computing time.

The main idea herein discussed is to reduce the problem's dimensions by avoiding discrete variables. The proposed formulation employs complementarity constraints to handle assignment and sequence decisions, applied on continuous variables. An NLP feasible point is equivalent to an MILP feasible point and vice versa. Therefore, an NLP local solution is equivalent to an integral MILP feasible point, defining an upper bound (if solving a minimization problem) on the correspondent MILP, which can improve the pruning in a branch-and-bound procedure. In fact, the NLP solution is a valid solution for the scheduling problem, and may be kept as the solution in a real-world situation or be used as an incumbent for the MILP.

In addition to the complementarity approach, we can also consider scheduling systems as dynamic systems, where one action (decision) at a given instant impacts the future states of the system. Dynamic systems are classically made up by control variables (the decisions one can make), state variables (the system features one can measure), and state equations (how a state is affected by past states and decisions). In industries such as the Oil and Gas industry and the Water and Wastewater industry, control systems are built upon optimal control dynamic models, where one tries to maintain the system operating safely and efficiently. An optimal control problem features a highly separable Jacobian of the constraints, with a block-diagonal structure, which may result in convergence with lower computational costs (Faco, 1990). Common NLP solvers can take advantage of this particular structure, as a discrete optimal control problem is equivalent to a NLP problem (Abadie, 1970). In particular, a scheduling problem can be exactly described with this approach: the transfer operations are represented by control variables, while inventories are mapped to state variables. The state of a given time instant is calculated from previous state and control variables, by means of the state equations.

In this paper, we combine both nonlinear approaches: complementarity and optimal control to avoid mixed-integer formulations. We use the scheduling of crude oil and derivatives in ports as an example for the proposed nonlinear optimal control model. The paper is divided as follows: Section 2 discusses the crude oil problem and its models; Section 3 presents some numerical examples; and Section 4 closes this work with our final remarks.

2. Crude Oil Scheduling Models

The scheduling of crude oil and derivatives in ports is the problem to determine: (i) ship allocation within the port; (ii) transfer operations between ships, tanks, process units, and

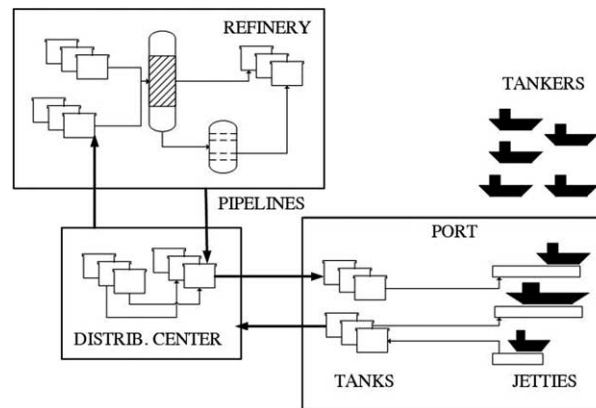


Fig. 1. Logistic subsystems.

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. It is a complex task, featuring nonlinear (due to crude blending) and combinatorial (due to assignment and sequencing) aspects.

The logistic system can be divided in three main subsystems (Fig. 1): port, distribution center, and refinery, all of them connected by pipelines (Mas and Pinto, 2003). It is also possible to consider a single system, when the port tanks are directly connected to refinery charging tanks (Shah, 1996; Magalhaes and Shah, 2003). In addition to these three systems, one may possibly consider a fourth system: the tanker fleet, whose schedule updates the estimated times of arrival (ETA) for each ship. According to Shah (1996), a reasonable approach is to solve the systems hierarchically. We follow this approach in this paper, considering two systems: the port (tankers, jetties, tanks, and pipelines) and the refinery crude area (pipelines, tanks, and distillation crude unit). However, it is important to mention that the equations presented herein could be employed in other arrangements as well.

Portside tanks serve as a buffer to keep the pipelines in continuous operation, even when tankers are late. In general, a (refinery or portside) tank stores a certain class of crude (e.g., heavy oil tanks cannot store light oil). Ideally, a good schedule will 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 the refinery production plan or delaying the ships. The refinery's demand for crude oil (as well as derivatives production) must be met by the port scheduling. Jetties can be restrictive on what tankers and cargoes to handle, according to their dimensions (draught and length) and pumping capacity. A ship must berth, unload, and leave the port within a time window defined by contract, otherwise the oil company will pay heavy demurrage fees. For instance, Brazilian demurrage costs amounted to USD 1.5 billion in 2006 (Collyer, 2006).

Therefore, the port schedule's main objective is to minimize demurrage costs, while keeping the refinery plan. A jetty is available for berthing only after the previous ship had enough time to leave the port. In the refinery side, the crude distillation unit operates

continuously, around an operational feed flow. Blending is not allowed in the lines, i.e., each transfer operation has only one source equipment and one destination equipment at a given time. Running tanks are not allowed either, i.e., a tank cannot receive and send crude simultaneously. In fact, a tank can make a delivery to another equipment (e.g., pipeline or crude distillation unit) only if the necessary “idle time” has been observed (e.g., to separate brine from crude oil or to assure a lab analysis).

In the recent literature, Shah (1996) proposed a MILP formulation for crude oil scheduling from tanker vessels to CDUs, based on two models: (i) a refinery problem (called the downstream problem); and (ii) a port model (called the upstream problem), constrained by the pipeline parcels defined by the solution of the first problem. Magalhaes and Shah (2003) revisited the problem, extending Shah’s original MILP formulation to consider a real-world port–pipeline–refinery infrastructure and additional operational constraints. The authors pointed out that some optimal solutions of the MILP model, if applied to a real-world schedule, could be considered by a human scheduler as non-optimal, or even unfeasible, because certain real-world decisions are sometimes very hard to be mathematically modeled. Más and Pinto (2003) modeled another real-world infrastructure, dividing the crude oil logistic system in three subsystems: (1) port, (2) distribution centers (intermediate storage), and (3) refineries. They also presented an exponential equation to calculate an upper bound of binary variables with the number of time intervals in order to illustrate that real-world instances are hard to be solved.

2.1. Modeling the Transfer Operation

The fundamental scheduling activity is the transfer operation, which is made up by a pair of equipments (source–destination) connected by an arc, and a flow from the source to the destination. The control vector $u(t_i)$ is the vector whose each entry $u_j(t_i)$ stands for a nonnegative flow on arc j at time t_i . The optimization problem is to define a feasible sequence of $u(t_i)$, for all instants t_i , which minimizes the objective function J . All control variables $u_j(t_i)$ are bounded.

The infrastructure of a logistic system can be seen as a graph, defined during the problem’s formulation, featuring equipments as nodes, connected by flow arcs. A system graph is built by checking which equipments are connected by pump lines, and which are compatible in terms of crude oil and physical dimensions. Fig. 2 illustrates a port infrastructure with 3 jetties, 5 tanks (3 for end products, 2 for crude oils), 2 pipelines connecting the port to a refinery (one to receive end products, the other to send crude oil), and 3 tankers that must be scheduled. In this example, tanker N3 can berth on jetties P3 and P2, but cannot berth on jetty P1. Moreover, N3’s cargo is a crude oil that can be pumped to tank T5. Therefore, there is a flow arc (represented by the lower traced line) between tanker N3 and tank T5, through jetty P3.

In summary, the schedule is basically to define a nonnegative flow $u_j(t_i)$ for each arc j at each time instant t_i . If $u_j(t_i) = 0$, there is no transfer operation at arc j at time t_i , otherwise, a transfer operation is occurring at this arc.

Operational constraints, such as “one equipment N cannot be the destination of two transfer operations at the same time t_i , in order to avoid inline blending” can be modeled

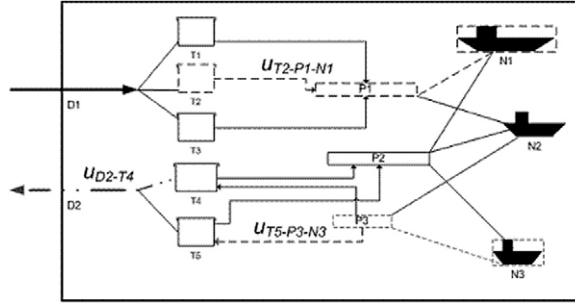


Fig. 2. System as a graph.

by different manners. For instance, let's examine the case of a certain tank N which is being fed by other equipments: it can be the destination of at most one transfer operation at a certain time t_i , as inline blending is forbidden. We present two different modeling possibilities (Table 1), with A_N as the set of indexes for all arcs whose destination is N : (a) an MILP formulation, as commonly found in the literature (Shah, 1996; Mas and Pinto, 2003), and (b) a novel NLP formulation proposed here.

The volume of tank N is calculated by a volumetric balance equation, which is equal in both models. The models differ on how to enforce the upper bounds on the flows and how to guarantee that only one source will be employed to feed the tank. Notice that, 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). Both formulations enforce this behavior.

Model (a) requires an additional control vector $b(t_i)$ of binary variables, whose each entry $b_j(t_i)$ is associated to the $u_j(t_i)$ entry. If $b_j(t_i)$ is set to 1, then a positive flow is allowed on arc j ; otherwise (if set to zero), no flow is allowed on arc j . This is assured by the manipulation of the bounds on $u_j(t_i)$: if $b_j(t_i) = 1$, the bounds are preserved; other-

Table 1
Modeling possibilities for transfer operations

Model	Equations
(a) MILP	$vol_N(t_i) = vol_N(t_{i-1}) + \sum_{j \in A_N} u_j(t_i) \Delta t$ $\sum_{j \in A_N} b_j(t_i) \leq 1$ $0 \leq u_j(t_i) \leq b_j(t_i) * u_j^{MAX}(t_i), j \in A_N$ $b_j(t_i) \text{ is binary, } u_j(t_i) \in R$
(b) NLP	$vol_N(t_i) = vol_N(t_{i-1}) + \sum_{j \in A_N} u_j(t_i) \Delta t$ $\sum_{j \in A_N} \sum_{k > j \in A_N} u_j(t_i) * u_k(t_i) = 0$ $0 \leq u_j(t_i) \leq u_j^{MAX}(t_i), j \in A_N$ $u_j(t_i) \in R$

wise, they are set to zero. The summation constraint on the binary constraints guarantees that at most one $b_j(t_i)$ can be evaluated as 1 at t_i , $j \in A_N$. All others binary variables associated to A_N must be evaluated as zero.

Model (b) relies on the control vector $u(t_i)$ 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 model. In the next sections, “idle time” and “berthing” constraints are formulated in a similar fashion.

2.2. Optimal Control Nonlinear Model

In the previous section, we have implicitly defined the schedule as a dynamic model within the optimal control framework.

DEFINITION 1. An optimal control problem is defined mathematically as the following Mathematical Programming problem. The control variables are u , the state variables are x and the time horizon spreads from t_0 to t_F .

Minimize:

$$J = f(u, x, t)$$

subject to:

$$\begin{aligned} u^{MIN} &\leq u(t_i) \leq u^{MAX}, & t_0 &\leq t_i < t_F, \\ x^{MIN} &\leq x(t_i) \leq x^{MAX}, & t_0 &\leq t_i \leq t_F, \\ x(t_i) &= g(x(t_{i-1}), u(t_{i-1})), & t_0 &< t_i \leq t_F, & x \in R^n, u \in R^m. \end{aligned}$$

The transfer operation is composed by two equipments – source and destination – and a flow u_j from the source to the destination (through arc j). A storage is filled by flows from other equipments, changing its state (volume and composition), and this equipment may later perform an outlet transfer operation performing changes on other equipments. We modeled flow rates as control variables, bounded by upper and lower bounds: each entry of the control-vector $u(t_i)$ stands for a flow in an arc between two equipments during the interval $[t_i, t_i + \Delta t)$. The control’s upper bounds may not be the same for all intervals, depending on port unavailability because of tides, limited operation of some equipments during certain periods of the day, scheduled maintenance, and the fact that a ship can only berth after its ETA. However, it is important to notice that all flow bounds are known *a priori*, during the problem formulation phase.

The proposed nonlinear optimal control model features the flow rates u as control variables (2), and volumes v and qualities p as subsets of the state variables x (3), all bounded by upper and lower limits. The state equations are developed from volume balance (4) and product blending in storage equipments (5). The objective-function (1) is a

summation of different costs, which can be prioritized with the use of weights (w_{cost}).

$$\text{Minimize } J = \sum_{cost} w_{cost} * C_{cost}, \quad (1)$$

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

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

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

$$p_{N,q}(t_i) = \left(v_N(t_{i-1}) * p_{N,q}(t_{i-1}) + \sum_{j \in A_N} u_j(t_{i-1}) * p_{j,q}(t_{i-1}) * \Delta t \right) / v_N(t_i). \quad (5)$$

Eq. (4) features U as an incidence square matrix with entries in $\{0, 1, -1\}$. Eq. (5) calculates density, sulphur concentration and product composition: each $u_j(t_{i-1})$ is an inlet flow at N at time t_{i-1} and $p_{j,q}(t_{i-1})$ is the value of this inlet flow's property q . The complementarity equations will force that at most one source equipment is actually feeding N , i.e., at most one $u_j(t_{i-1})$ will be greater than zero.

The following equations model scheduling decisions: unique definition of source and destination in a transfer operation (6), idle time to segregate impurities (7), berthing time (8), and constant flow constraints (9), and all of them must equal to zero. We define new state variables r_N , z_N , s_N , and q_N for all equipments N , each one referring to a complementarity equation. In the next section, these variables will be employed to relax and penalize the problem.

$$r_N(t_i) = \sum_{j \in A_N} \sum_{k > j \in A_N} u_j(t_{i-1})u_k(t_{i-1}) = 0, \quad (6)$$

$$z_N(t_i) = \sum_{t' = t_{i-1} - \Delta t_N^{IDLE}}^{t_{i-1}} \sum_{j \in A_N} \sum_{k \in A_N^{OUTLET}} u_j(t')u_k(t_{i-1}) = 0, \quad (7)$$

$$s_N(t_i) = \sum_{t' = t_{i-1} - \Delta t_N^{BERTH}}^{t_{i-1}} \sum_{j \in A_N} \sum_{k \in A_{K < > N}} u_j(t')u_k(t_{i-1}) = 0, \quad (8)$$

$$q_N(t_i) = u_{N,0} - \sum_{j \in A_N} u_j(t_i) = 0. \quad (9)$$

Eq. (6) enforces that only one flow can feed N at time t_{i-1} , therefore, any transfer operation has only one source and only one destination at time t_{i-1} . Eq. (7) enforces that N will be able to feed another equipment only after its idle time Δt_N^{IDLE} was respected. Eq. (8) enforces the necessary berthing time Δt_N^{BERTH} for ships. Eq. (9) force a constant flow $u_{N,0}$ feeding a given equipment (usually a process unit or a pipeline) N – this constraint can be easily changed to force a variable flow, if needed.

In the case of crude oil scheduling, we considered the following costs: demurrage (Eqs. (10)–(12)), unattained demand (13), and inventory (14).

$$C_{demurrage}^{unload} = \sum_{N \in Ships^{unload}} \sum_{t_i > t_N^{depart}} c_N^{demur} v_N(t_i), \quad (10)$$

$$C_{demurrage}^{load} = \sum_{N \in Ships^{load}} \sum_{t_i > t_N^{depart}} c_N^{demur} (Cargo_N - v_N(t_i)), \quad (11)$$

$$C_{demurrage} = C_{demurrage}^{load} + C_{demurrage}^{unload}, \quad (12)$$

$$C_{demand} = \sum_{t_i} \sum_{N \in Pipelines} \sum_{P \in Products} c_N^{demand} v_{N,P}(t_i), \quad (13)$$

$$C_{inventory} = \sum_{t_i} \sum_{N \in Storages} c_N^{inv} v_N(t_i). \quad (14)$$

Eqs. (10), (11), and (12) deal 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. Notice that demurrage costs are accounted for ship N only if it has departed after the maximum acceptable time of departure (t_N^{depart}). If ship N is to be loaded, its volume at the departure time must be $Cargo_N$. If N is to be unloaded, its volume at the departure time must be zero.

In all cost equations, c^{cost} is a different arbitrary unitary cost.

2.3. Solving the Problem

The model is solved as follows: the nonlinear constraints (6)–(8) are relaxed and added to the objective-function as penalties, creating the merit-function J' (15). This merit-function will be minimized instead of the original objective-function J . This approach removes most of the difficult constraints, generating a broader search region for nonlinear optimization methods, with fewer constraints. In addition, if only linear state equations are present, the search region becomes a polyhedra. Within the feasible region of the original formulation, all penalties are cancelled. The parameter μ can be determined iteratively by solving successive relaxations of the original problems or fixed *a priori* as a large enough number.

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

where e is the unary-vector.

Trivial points are points where the control-vector u is zero for all time intervals. These points are very easy to be constructed, but they are not feasible in the original problem formulation. At trivial points, the demurrage and demand costs are maximal. The norms of the additional states are $\|z\| = \|r\| = \|s\| = 0$ and $\|q\| \gg 0$. However, such points are feasible in the relaxed formulation and define a descent direction that leads to the minimization of the penalties – moving the points to the feasible region. Therefore, we use these points as starting points.

2.4. Mixed Integer Linear Model

In order to compare the NLP approach with the more common MILP approach, we present an MILP model following the crude oil scheduling literature (Shah, 1996) and

(Mas and Pinto, 2003). Eqs. (2), (6)–(8) from the NLP model are replaced by Eqs. (16)–(19). The blending equation (5) is dropped, as it is nonlinear. Complementarity constraints are replaced by mixed-integer constraints, with the addition of the binary variables vector (b_j).

$$0 \leq u_N(t_i) \leq \text{Diag}(b_{j \in A_N}(t_i)) u_N^{MAX}(t_i), \quad (16)$$

$$\sum_{j \in A_N} b_j(t_i) \leq 1, \quad (17)$$

$$\sum_{j \in A_N^{OUTLET}} b_j(t_i) + \sum_{t'=t_i-\Delta t^{BERTH}}^{t_i} \sum_{j \in A_N} b_j(t') \leq 1, \quad N \in Ships, \quad (18)$$

$$\sum_{j \in A_N^{OUTLET}} b_j(t_i) + \sum_{t'=t_i-\Delta t^{IDLE}}^{t_i} \sum_{j \in A_N} b_j(t') \leq 1, \quad N \in Storages. \quad (19)$$

Eq. (16) features a diagonal matrix $\text{Diag}(b_j(t_i))$, 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 $b_j = 1$. The consecutive equations represent the following constraints: only one flow can be used by an equipment N at time t_{i-1} , idle time and berthing time must be respected before any outlet transfer.

The MILP can be solved with usual mixed-integer procedures and is larger than the NLP model, as shown in next section.

3. Results

The NLP and MILP models were compared in 5 preliminary test instances (Tables 2 and 3), coded in AMPL (Fourer *et al.*, 2003), and solved with standard commercial solvers: CPLEX (v. 10.1.0) (ILOG, 2002), SNOPT (v. 6.1; Gill *et al.*, 2002), and MINOS (v. 5.5) (Murtagh and Saunders, 1982). Case 1 is composed by an infrastructure with two crude tanks and one pipeline connected to a refinery, whose crude demand has to be fulfilled. Case 1 has 2 configurations: (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. Case 2 has two crude tanks, one jetty and two tankers, whose cargo had to be unloaded. Case 3 has three crude tanks, one jetty, three tankers, whose cargo had to be unloaded, and one pipeline, whose demand has to be fulfilled. Case 3 has 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 number of variables is shown as determined after AMPL's pre-solve procedure. As both MILP and NLP models have linear objective functions, it is possible to compare them in regards of the global optimality of their solutions. All cases were solved in a workstation with the following

Table 2
Dimensions of the test cases

Model	Case	Binary variables	Continuous variables	Constraints
NLP	1(A)	0	31	25
	1(B)	0	31	30
	2	0	111	87
	3(A)	0	169	103
	3(B)	0	135	97
MILP	1(A)	12	25	31
	1(B)	12	25	36
	2	34	82	111
	3(A)	93	158	265
	3(B)	93	158	275

Table 3
Results

Model	Case	Solution	Iterations	Global optimum
NLP (SNOPT)	1(A)	1460	51	Yes
	1(B)	1600	13	Yes
	2	0.33	12	Yes
	3(A)	0	812	Yes
	3(B)	18.27	544	No
NLP (MINOS)	1(A)	1460	17	Yes
	1(B)	1625	5	No
	2	0.33	191	Yes
	3(A)	0	411	Yes
	3(B)	0	472	Yes
MILP (CPLEX)	1(A)	1460	14	Yes
	1(B)	1600	13	Yes
	2	0.33	63	Yes
	3(A)	0	324 (8 BB nodes)	Yes
	3(B)	0	397 (25 BB nodes)	Yes

configuration: Intel Core Duo T2250 1.73GHz, RAM 1 GB, Linux OpenSUSE 10.1. The running times were around 1s.

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 equiva-

lent to an MILP feasible point, we propose a hybrid scheme: 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 branch-and-bound tree. This scheme can 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. At the current state of our research, we employed the NLP solutions to initialize the MILP previous examples, 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). All cases had similar CPU times of approximately 1s.

Table 4 shows the MILP iterations when the NLP solutions were employed as initial incumbents to the MILP problem. A solution defined by the NLP formulation is transformed to a 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 null flow, the corresponding binary variable is set to 0 (zero). The number of branch-and-bound iterations and visited nodes is significantly reduced, even for these preliminary test cases.

4. Solving a Larger Problem

In addition to the preliminary cases, we solved a larger problem (Case 4) from the literature (Moro and Pinto, 2004) with the proposed NLP approach, where one has to schedule

Table 4
MILP results with different initializations

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

the receipt of 4 crude cargoes with 3 different crude oils within 6 crude tanks in a refinery over 5 days, keeping the refinery's crude processing unit (CDU) continuously running. The problem is similar to Case 3(B): for scheduling purposes, the CDU is identical to a pipeline receiving crude from the tanks. Moro and Pinto proposed two models: an MILP and an MINLP. The MINLP employed a nonlinear blending equation to calculate crude oil composition (similar to Eq. (5)) and the MILP employed a linear approximation of this equation. Both models had the objective of processing the maximum amount of crude oil as possible. In their paper, Moro and Pinto reported that they were able to achieve such objective at the MINLP solution, whereas their MILP was stopped in a solution that could not process all crude (the CDU operated under its maximum flow capacity).

Tables 5 and 6 compare our NLP with Moro and Pinto's MILP and MINLP as reported in their paper: MILP solved with CPLEX (ILOG, 2002) in 105 min and MINLP solved with DICOPT (Viswanathan and Grossmann, 1990) in 30 min, both in a 700 MHz Pentium PC. Our NLP was solved with SNOPT (Gill *et al.*, 2002) at the NEOS Server (Dolan *et al.*, 2002) in 30 s (job 1701755 at host shepherd.mcs.anl.gov). It is important to state that the computational times cannot be directly compared as the configuration of the NEOS server is unknown to us. However, the small number of iterations is a strong indicative that the NLP approach is a valid alternative to solve this kind of problem.

5. Conclusion

A nonlinear optimal control model for process scheduling – based on flow variables – was presented. All constraints are modeled without discrete variables, achieving a continuous model. These models are smaller than their MILP counterparts from the literature. Although being capable of generating good solutions, the NLP formulation is nonconvex.

Table 5
Comparison of model dimensions

Model	Case	Binary variables	Continuous variables	Constraints
NLP	4	0	3727	3108
MILP	4	912	4246	5175
MINLP	4	228	2226	2634

Table 6
Results

Model	Case	Solution (processed crude in m ³)	Iterations
NLP (SNOPT)	4	168000	33536
MILP (CPLEX)	4	143109	856297
MINLP (DICOPT)	4	168000	330370

The NLP can be employed as an auxiliary problem to traditional MILP formulations. In fact, preliminary numerical results showed a significant reduction of MILP iterations when initialized by a NLP solution.

References

- Abadie, J. (1970). Application of the GRG algorithm to optimal control problems. In J. Abadie (Ed.), *Integer and Nonlinear Programming*, North-Holland.
- Al-Yakoob, S.M., and H.D. Sherali (2007). Multiple shift scheduling of hierarchical workforce with multiple work centers. *Informatica*, **18**(3), 325–342.
- Collyer, W. (2006). Sobreestadia de navios: a regra 'once on demurrage, always on demurrage'. *Jus Navigandi*, 1166 (in Portuguese).
<http://jus2.uol.com.br/doutrina/texto.asp?id=8889> (last access on July, 2008).
- Dzemyda, G. (1999). Mean squared load criteria for scheduling independent tasks. *International Journal of Applied Mathematics and Computer Science*, **9**(4), 101–116.
- Dolan, E.D., R. Fourer, J.J. More and T.S. Munson (2002). The NEOS server for optimization: Version 4 and beyond. *Mathematics and Computer Science Division*, Argonne National Laboratory.
ftp://info.mcs.anl.gov/pub/tech_reports/reports/P947.pdf (last access on November, 2008).
- Faco, J.L.D. (1990). A generalized reduced gradient algorithm for solving large-scale discrete-time nonlinear optimal control problems. In H.B. Siguerdjane and P. Bernhard (Eds.), *Control Applications of Nonlinear programming and Optimization*. Pergamon Press, Oxford.
- Floudas, C.A., and X. Lin (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers and Chemical Engineering*, **28**, 2109–2129.
- Floudas, C.A., and X. Lin (2005). Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Annals of Operations Research*, **139**, 131–162.
- Fourer, R., D.M. Gay and B.W. Kernighan (2003). *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press.
- Gill, P.E., W. Murray and M.A. Saunders (2002). SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Journal on Optimization*, **12**(4), 979–1006.
- ILOG, Inc. (2002). *ILOG AMPL/CPLEX System Version 8.0 User's Guide*. New York.
- Janiak, A., and M.Y. Kovalyov (2006). Job sequencing with exponential functions of processing times. *Informatica*, **17**(1), 13–24.
- Más, R., and J.M. Pinto (2003). A mixed-integer optimization strategy for oil supply in distribution complexes. *Optimization and Engineering*, **4**(1), 23–64.
- Magalhaes, M.V., and N. Shah (2003). Crude oil scheduling. In *Proceedings of the 4th Conference on Foundations of Computer-Aided Process Operations*. pp. 323–326.
- Moro, L.F.L., and J.M. Pinto (2004). Mixed-integer programming approach for short-term crude oil scheduling. *Industrial Engineering and Chemistry Research*, **43**, 85–94.
- Murtagh, B.A., and M.A. Saunders (1982). A projected Lagrangian algorithm and its implementation for sparse non-linear constraints. *Mathematical Programming Studies*, **16**, 84–117.
- Shah, N. (1996). Mathematical programming techniques for crude oil scheduling. *Computers and Chemical Engineering*, **20**(suppl.), S1227–S1232.
- Viswanathan, J., and I.E. Grossmann (1990). A combined penalty function and outer approximation method for MINLP optimization. *Computers and Chemical Engineering*, **14**, 762–775.

F.D. Fagundez is a chemical engineer whose main research interests are modeling and optimization of logistic systems with mathematical programming, especially in the oil and gas industry. His current PhD research focuses on the use of novel nonlinear programming approaches to solve scheduling problems. He currently works at BNDES, the Brazilian Development Bank.

A.E. Xavier is a professor at the Systems Engineering and Computer Science Program of COPPE, the Engineering Graduate School of the Federal University of Rio de Janeiro. His main research interests are on the optimization of clustering, energy production and hydrological problems with penalty-based NLP methods, in particular the hyperbolic penalty method.

J.L.D. Faco is a professor at the Department of Computer Science of the Institute of Mathematics of the Federal University of Rio de Janeiro. His main research interests are on optimal control, simulation, optimization and modeling of real problems on logistics, energy production and waste treatment, in particular with the use of methods from the generalized reduced gradient family.

Tolydaus netiesinio programavimo metodai tvarkaraščių problemoms spręsti

Fabio D. FAGUNDEZ, Adilson E. XAVIER, Joao L.D. FACO

Tvarkaraščių sudarymas yra bene žinomiausias diskrečiojo optimizavimo uždavinys. Kita vertus, gerai žinomi tolydaus optimizavimo pritaikymai yra susiję su dinaminių sistemų valdymu. Šiame darbe yra derinami abu būdai, sprendžiant tvarkaraščių planavimo uždavinius kaip diskretaus laiko netiesinio dinaminio valdymo problemą su būsenos ir valdymo kintamaisiais, atitinkančiais darbų atlikimo viršutines ir apatines ribas. Papildomumo sąlygos panaudojamos tvarkaraščiams pavaizduoti. Uostų darbų planavimo pavyzdys yra detaliai nagrinėjamas pateikiant skaitmeninius rezultatus.