The Offshore Resources Scheduling Problem: Detailing a Constraint Programming Approach

Thiago Serra, Gilberto Nishioka, and Fernando J.M. Marcellino

PETROBRAS – Petróleo Brasileiro S.A.
Avenida Paulista, 901, 01311-100, São Paulo – SP, Brazil
{thiago.serra,nishioka,fmarcellino}@petrobras.com.br

Abstract. The development of maritime oil wells depends on the availability of specialized fleet capable of performing the required activities. In addition, the exploitation of each well can only start when it is connected through pipes to a producing unit. The Offshore Resources Scheduling Problem (ORSP) combines such restrictions with the aim to prioritize development projects of higher return in oil production. In this work, an extended description of the ORSP is tackled with Constraint Programming (CP). Such extension refers to the customization of pipes and to the scheduling of fleet maintenance periods. The use of CP is due to the noted ability of the technique to model and solve large and complex scheduling problems. For the type of scenarios faced by Petrobras, the reported approach was able to quickly deliver good-quality solutions.

Keywords: Scheduling, Time-interval Variables, Well Developments

1 Introduction

As Daniel Yergin observed, "the competition for oil and the struggle for energy security seem to never end" [24]. More than 80% of the world primary energy consumption is currently supplied by fossil fuels. Among them, oil is considered the scarcest source because it has the smallest reserves-to-production ratio [6]. Once an opportunity is detected, the exploitation of a new oil reservoir usually requires large investments, and sometimes incurs a number of technological challenges. Such is the case currently faced by Petrobras with the Tupi field, which is located in the pre-salt layer along the Brazilian coast. Announced in 2007 as the largest oil discovery of the last 30 years in the Western hemisphere, it was estimated to contain from 5 to 8 billion recoverable barrels of oil equivalent [7]. On account of enabling its exploitation, Petrobras had the largest public share offering ever seen [5]. The company now possesses a 5-year investment budget of 225 billion US dollars [20], which is mainly focused on exploiting the pre-salt layer. In this context, the optimized use of resources is critical to leverage the success of such ventures and to help attending the increasing demand of the society for energy.

One of the challenges to exploit new oil fields consists of mitigating the shortage of machinery. Resources such as a specialized fleet are highly demanded but also scarce and very expensive. That is the case of oil platforms – usually referred as oil rigs – to drill and complete wells, and pipelay vessels to carry pipes to connect developed wells to producing units. The use of such resources has been approached in a variety of ways along the life cycle of oil reservoirs. For instance, the scheduling of exploratory activities for preliminary assessment was addressed by Glinz and Berumen [8]; the integrated planning and scheduling of well developments in oil fields was approached by Iyer et al. [12]; and the scheduling of well maintenance activities was focused by Aloise et al. [2]. The development of wells consists of a midpoint between the latter two cases. It is possible to address it by means of the Offshore Resources Scheduling Problem (ORSP), which is aimed at prioritizing developments with higher combined return of oil production.

Hasle et al. [9] introduced this problem, which was shown to be NP-hard by Nascimento [18]. Since then, many approaches have been refining its extent and solving methodology [19, 17, 21–23]. The decisions consist of selecting well development projects, and then assigning and sequencing resources to perform each of their activities. The optimization criterion is to maximize the incurred short-term production of oil. In addition, its constraints avoid the assignment of an unfit resource to any activity, provide time for routine maintenance of those resources, and limit their presence at any site to a maximum allowed for security reasons. Hence, solving the ORSP can aid the operational decision-making in a process that is directly related to the return on investment of oil companies.

There is a long record of approaches to the ORSP at Petrobras. Most of it is related to a system named ORCA, which stands for Optimization of Resources of Critical use in Activities for exploitation. Its development started with the work of Accioly et al. [1], which focused mostly on well maintenance activities. Their motivation was the fact that fewer resources are usually available for such type of activity, thus creating a demand-pull for the use of optimization. Pereira et al. [19] and Moura et al. [17] broadened the focus of the system to well developments. More recently, new efforts have been reported by Serra et al. [21–23] to tackle larger instances and to cope with new bottlenecks that have been recognized. Resources regarded as critical to well developments were split between rigs and vessels, and the inventory of pipes on vessels was included in [21]. The model introduced in [21] was refined and extended in [23], when the selection of development projects became a problem decision. The first attempt to provide an upper bound to the production of each instance was made in [21] and then tightened in [22]. Each of those efforts has contributed to improve the quality of the achieved solutions and to reduce their gap with respect to operational needs.

This work reports an extended description of the ORSP as well as a model to tackle it using Constraint Programming (CP). The current refinement refers to the remaining details that were included after the development of ORCA was resumed. They are related to the availability of specific pipes for connecting each well, and to the scheduling of resource maintenance activities. We aim to present a detailed description of the developed CP model and to evaluate its performance on a benchmark of instances based on a past scenario of the company.

The paper is organized as follows. Section 2 contains a definition of the ORSP. The rationale for using CP is presented in section 3, while section 4 describes the developed model and its evaluation. The benefits attributed to CP are discussed in section 5. Final remarks are presented afterwards.

2 The Offshore Resources Scheduling Problem

The ORSP consists of scheduling oil rigs and pipelay vessels to develop oil wells while targeting to maximize the incurred short-term production of oil. Rigs and vessels are required to attend to a large geographical area, in which displacements may take a considerable time. They are scheduled to perform development activities at wells, auxiliary loading activities at harbors and their own maintenance. Well development and loading activities must be assigned to a resource compatible with their needs, thus comprising requirements such as if the rig is able to operate at the depth of a given well or if the vessel is able to unload at once a given weight of pipes. Some of the development activities require pipes which, in turn, must have been previously loaded at harbors; and the amount of loading activities varies according to how many pipes are loaded at each time.

The problem decisions are *if*, *when* and *how* to perform each of those activities. Time is represented in days, starting from a date set as 0. The following convention is used to describe the problem. Resources will be denoted by index $\mathbf{i}$ assuming values in the set $\mathbf{I}$, where $\mathbf{I_R} \subseteq I$ denotes the set of rigs and $\mathbf{I_V} = I \setminus I_R$ the set of vessels. Activities will be denoted by index $\mathbf{j}$ assuming values in the set $\mathbf{J}$, where $\mathbf{J_W}$ denotes the set of well development activities, $\mathbf{J_H}$ the set of loading activities, $\mathbf{J_M}$ the set of resource maintenance activities, $J_W \cup J_H \cup J_M = J$ and these subsets are pairwise disjoint. Let $\mathbf{J_{M'}} \subseteq J_M$ be the set of resource maintenances allowing concurrency of other activities. The compatibility of assignments is represented by matrix $\mathbf{C}$, where $c_{ij} = 1$ if, and only if, resource i is compatible with activity j. Locations will be denoted by index $\mathbf{k}$ assuming values in the set $\mathbf{K}$, where $\mathbf{K_W} \subseteq K$ denotes the set of wells and $\mathbf{K_H} = K \setminus K_W$ the set of harbors. Pipes will be denoted by index $\mathbf{p}$ assuming values in the set $\mathbf{P}$.

2.1 Optimization criterion

The short-term production is measured as how much each developed well would produce until a time horizon $\mathbf{H}$. The conclusion of each activity j induces a daily production rate $\mathbf{pr_j}$, which is nonzero only if it finishes a well development.

2.2 Resource constraints

- Each resource performs at most one development or load at a time.
- Only one resource can be placed at a well at any time.
- Each harbor k supports up to $\mathbf{s_k}$ resources at the same time.
- If a resource i performs consecutive activities on distinct locations k_1 and k_2, there must be a minimum displacement time $\mathbf{dt_{ik_1k_2}}$ between them.
- Each resource i has a contractual period of use, ranging from the release date $\mathbf{rr_i} \geq 0$ to the deadline $\mathbf{rd_i}$.

2.3 Activity constraints

- Activities are non-preemptive, i.e., they are performed without interruption.
- Each development or maintenance j can only be scheduled between release date $\mathbf{ar_j} \geq 0$ and deadline $\mathbf{ad_j} \geq ar_j$. It is mandatory if $ad_j < H$ or $j \in J_M$.
- Each activity j is associated with a location $\mathbf{loc_j}$.
- Either all activities of a well are performed or none is.
- Each development or maintenance j requires $\mathbf{p_j}$ days to be performed.
- The length of a loading activity on resource i ranges from $\mathbf{mil_i}$ to $\mathbf{mal_i}$ days.
- If a well development activity j_2 must be preceded by another activity j_1, then $\mathbf{pc_{j_1 j_2}} = 1$, and $\mathbf{pd_{j_1 j_2}}$ denotes the minimum delay between both.
- Each well development activity j belongs to a cluster of activities with index $\mathbf{cl_j}$, and all activities of a cluster must be assigned to a single resource.
- Each resource maintenance activity j is associated with a resource $\mathbf{rm_j}$.
- For each maintenance activity $j \in J_{M'}$, which only makes a resource partially unavailable, let $\mathbf{J'_j} \subseteq J_W$ be the set of activities that cannot concur with it.

2.4 Inventory constraints

- Only vessels may perform loading activities at harbors.
- Each vessel has an empty inventory at time 0.
- Each vessel i has an inventory varying between 0 and $\mathbf{ic_i}$ along time.
- Each loading activity takes one or more pipes at a harbor.
- The number of loading activities is variable. However, an upper limit can be set with as much activities at each harbor as the number of pipes it has.
- A pipe p is only loaded at harbor $\mathbf{hp_p}$ from the release date $\mathbf{rp_p}$ onwards.
- The load of a pipe p implies an inventory increase of $\mathbf{wp_p}$.
- The increase of the onboard inventory due to a loading activity is limited by the time it lasts, with such a rate that it can achieve ic_i after mal_i days.
- The unload of a pipe p is made by a development activity of connection $\mathbf{ca_p}$, which can only be assigned to a resource that has previously loaded pipe p.

3 Why CP

The selection of CP in the current project stems from a balance between performance and maintainability. Many stakeholders were leaning towards the use of a commercial solver in order to benefit from a modeling environment favoring fast prototyping. It was also seen as beneficial that such solvers use state-of-art algorithms that have been broadly tested elsewhere. Since many solvers support CP or Mathematical Programming (MP), there was a strong preference for either of those techniques. There was also a great concern with the timely deliver of good solutions. A successful use of CP was reported by early approaches both inside [1] and outside [9] the company. Metaheuristics were considered as well by Nascimento [18] and for a while at Petrobras due to Pereira et al. [19] and Moura et al. [17]. However, the favorable results of a metaheuristic against CP in these latter works are somewhat related to the fact that the tested instances were not much constrained. In particular, such prevalence was not as strong when large clusters of activities had to be assigned to a single resource. Due to the introduction of loading activities, higher levels of constrainedness were expected. In addition, the observed development of propagation mechanisms for resource constraints [3, 4, 13] favored using CP to model inventory on vessels. There was also some skepticism regarding the possibility of using MP due to the increase in the size of the expected instances. In such cases, CP solvers have a competitive edge because their models are more concise and feasible solutions are found earlier in the search. Hence, we pondered that CP was a reasonable choice.

Nevertheless, both CP and MP were employed in the developments that followed. To highlight the observed progress, we will use the results achieved with the largest available instance. In accordance to Hooker [10], equal hardware conditions were observed in all cases. The first solution was obtained by a CP model described in [21], where an ad-hoc algorithm assessed a worst-case optimality gap of 38%. Such gap was then reduced to almost 18% with a continuous-time Mixed-Integer Linear Programming (MILP) model without loading activities in [22]. Such experiment was helpful to assess the scalability issues of a straightforward MP approach to a scheduling problem: compared to CP, time and memory requirements increased at a much faster step. In the following, an improved CP model with project selection in [23] was able to find a solution 11% better. Thus, the worst-case optimality gap of the largest instance tested in previous works is below 6%. Such results increased our confidence on CP, which was selected to tackle the ORSP in the full extent currently required at Petrobras.

4 How CP

The ORSP was modeled with abstractions based on conditional time-interval variables. Such concept was introduced by Laborie and Rogerie [14] and further extended by Laborie et al. [15]. It specializes the abstract modeling targeted with the Optimization Programming Language (OPL) [16] to handle resource-constrained scheduling problems. For that sake, abstractions from the specialized CP subfield of Constraint-Based Scheduling (CBS) [3, 4] such as activities and resources are defined by means of intervals and other auxiliary elements.

Each interval variable depicts an event through a collection of interdependent properties such as its presence, start time, length, and end time. Such intervals can be used to model complex relations according to a hierarchical structure imposed by one-to-many constraints as well as sequence variables, cumulative and state functions, and the constraints that can be imposed on them. Further details about those elements are presented along the model description.

4.1 Modeling the ORSP

The model described in this section is an extension of the model M_2, which was introduced in [23]. It is depicted below with separate sections for the definition of variables and domains, constraints, objective function and search phases.

Variables and domains Interval variables are used to represent each of the problem activities in vector $\mathbf{a}$, and each combination of a resource and an activity requiring assignment in matrix $\mathbf{M}$. To each activity $j \in J$, there is a corresponding interval variable a_j in vector a. In the case of well development and resource maintenance activities, the following domain restrictions apply:

$$start(a_j) \geq ar_j, \quad \forall j \in J_W \cup J_M \quad (1)$$
$$end(a_j) \leq ad_j, \quad \forall j \in J_W \cup J_M \quad (2)$$
$$length(a_j) = p_j, \quad \forall j \in J_W \cup J_M \quad (3)$$
$$presence(a_j) = 1, \quad \forall j \in J_W, ad_j < H \quad (4)$$
$$presence(a_j) = 1, \quad \forall j \in J_M \quad (5)$$

Each cell m_{ij} of matrix M corresponds to the combination of resource $i \in I$ and activity $j \in J_W \cup J_H$. Thus, the presence of an interval m_{ij} implies the assignment of resource i to activity j. Since most of the resource-activity pairs are incompatible, the actual implementation of M is aimed to leverage such sparsity: separate tuple-indexed vectors are employed for activities on wells and activities on harbors; and each of them contains only intervals corresponding to compatible combinations, i.e., m_{ij} is there if, and only if, $c_{ij} = 1$. However, since such details are more of an implementation issue than a modeling design choice, the matrix notation was kept for simplicity. The following domain restrictions are imposed to intervals of M related to well developments:

$$start(m_{ij}) \geq MAX(rr_i, ar_j), \quad \forall i \in I, j \in J_W, c_{ij} = 1 \quad (6)$$
$$end(m_{ij}) \leq MIN(rd_i, ad_j), \quad \forall i \in I, j \in J_W, c_{ij} = 1 \quad (7)$$
$$presence(m_{ij}) = 0, \quad \forall i \in I, j \in J_W, c_{ij} \neq 1 \quad (8)$$

For loading activities, the corresponding cells have the following restrictions:

$$start(m_{ij}) \geq rr_i, \quad \forall i \in I, j \in J_H, c_{ij} = 1 \quad (9)$$
$$end(m_{ij}) \leq rd_i, \quad \forall i \in I, j \in J_H, c_{ij} = 1 \quad (10)$$
$$length(m_{ij}) \geq mil_i, \quad \forall i \in I, j \in J_H, c_{ij} = 1 \quad (11)$$
$$length(m_{ij}) \leq mal_i, \quad \forall i \in I, j \in J_H, c_{ij} = 1 \quad (12)$$

Since we have considered the existence of a loading activity for each pipe, we can refer explicitly to such correspondence with a bijection $\mathbf{f} : P \rightarrow J_H$ mapping each pipe $p \in P$ to a loading activity $j \in J_H$. Without loss of generality, we can use it to postpone the early start of each loading activity according to the release date of its corresponding pipe, and thus reduce the search space:

$$start(a_j) \geq rp_p, \quad \forall j \in J_H, p \in P, f(p) = j \quad (13)$$

Such premise also incurs that $loc_j = hp_p$, $\forall j \in J_H, p \in P, f(p) = j$, i.e., it associates each loading activity with the harbor of its corresponding pipe.

The use of each resource and the concurrency of resources on each location are represented by the vectors of cumulative functions $\mathbf{u}$ and $\mathbf{x}$, respectively.

Cumulative functions Cumulative functions are piecewise time-domain functions whose discrete changes in value are associated with the start and end of interval variables. They represent a generalization of cumulative resources [15]; for which many types and levels of consistency have been proposed and refined, and a number of filtering algorithms has been studied [3]. The composition of a cumulative function consists of a linear combination of steps, pulses and other cumulative functions. A step can be associated with the start and end of an interval with functions *stepAtStart* and *stepAtEnd*, respectively; and a pulse, which corresponds to equal but opposite steps at the start and at the end of an interval, is declared with function *pulse*. In either case, the first argument is an interval variable, which can be followed by one argument, a, if the variation is fixed or two arguments, a and b, if the variation must lay in the range $[a, b]$.

To each resource $i \in I$ there is an associated cumulative function u_i composed of unitary pulses associated with intervals of M, and to each location $k \in K$ there is a cumulative function x_k composed of unitary pulses on intervals of a:

$$u_i = \sum_{j \in J_W \cup J_H : c_{ij} = 1} pulse(m_{ij}, 1), \quad \forall i \in I \quad (14)$$

$$x_k = \sum_{j \in J_W \cup J_H : loc_j = k} pulse(a_j, 1), \quad \forall k \in K \quad (15)$$

The balance of inventory on the resources is modeled with the vector of cumulative functions $\mathbf{b}$. There is one cumulative function b_i corresponding to each vessel $i \in I_V$. It is composed of the sum of cumulative functions related to each resource-harbor pair in the matrix $\mathbf{BH}$. In turn, each cumulative function bh_{ik} from BH is composed of positive steps at the end of each loading activity performed by resource i at harbor k and negative steps for each activity on resource i that is associated with the release of a pipe from harbor k. The increase of inventory due to each load is variable, non-negative and limited to ic_i. The decrease of inventory is fixed and given by the weight of the unloaded pipes. Thus, we have the following definitions of the elements of BH and b:

$$\begin{aligned} bh_{ik} = & \sum_{j \in J_H : c_{ij} = 1 \wedge loc_j = k} stepAtEnd(m_{ij}, 0, ic_i) \\ & - \sum_{j \in J_W, p \in P : c_{ij} = 1 \wedge ca_p = j \wedge hp_p = k} stepAtEnd(m_{ij}, wp_p), \\ & \qquad \forall i \in I_V, k \in K_H \quad (16) \\ b_i = & \sum_{k \in K_H} bh_{ik}, \qquad \forall i \in I_V \quad (17) \end{aligned}$$

Finally, the location of the resources along time is represented with the vector of state functions $\mathbf{l}$. A state function represents a qualitative property that varies in time. It can be accompanied by an auxiliary function that defines the minimum transition time between its states. In the current case, let $d_i : K \times K \to \mathbb{N}$ be a function of resource $i \in I$ such that $d_i(k_1, k_2) = dt_{ik_1k_2}$, $\forall k_1 \in K, k_2 \in K, k_1 \neq k_2$. Thus, d_i corresponds to the displacement function of resource i. It can be used in combination with l_i to guarantee the minimum displacement time between activities on different locations that were assigned to resource i:

$$l_i : \mathbb{N} \rightarrow K \cup \{0\}, \qquad \forall i \in I \quad (18)$$

$$[l_i(t_1) = k_1 \wedge l_i(t_2) = k_2] \rightarrow t_1 + d_i(k_1, k_2) \leq t_2, \qquad \forall i \in I, k_1 \in K, k_2 \in K, k_1 \neq k_2, t_1 \in \mathbb{N}, t_2 \in \mathbb{N}, t_1 < t_2 \quad (19)$$

The value 0 represents the state at which the resource is not being used anywhere.

Constraints Vector a and matrix M are bound by constraint *alternative*, which is employed to state that at most one interval of the j-th column of M occurs and that it corresponds to interval a_j:

$$alternative(a_j, M_j), \qquad \forall j \in J_W \cup J_H \quad (20)$$

The clustering constraints are represented by logical implications, which force the presence of all development intervals of a cluster in a single line of M:

$$presence(m_{i_1j_1}) \rightarrow \neg\, presence(m_{i_2j_2}), \qquad \forall i_1 \in I, i_2 \in I, i_1 \neq i_2, j_1 \in J_W, j_2 \in J_W, j_1 \neq j_2, c_{i_1j_1} = 1, c_{i_2j_2} = 1, cl_{j_1} = cl_{j_2} \quad (21)$$

In order to prevent resources from being assigned to more than one activity at a time, an upper limit of 1 is set to the functions of u. Besides, the functions of l are set as equal to the location of the activities performed on each resource:

$$u_i \leq 1, \qquad \forall i \in I \quad (22)$$

$$[presence(m_{ij}) = 1] \rightarrow [l_i(t) = loc_j], \qquad \forall i \in I, j \in J_W \cup J_H, c_{ij} = 1, t \in [start(m_{ij}), end(m_{ij})) \quad (23)$$

With constraints on vector u, we can also prevent any resource from performing an activity during a period of full unavailability. That is made by constraining the value of the functions of u to 0 when such type of maintenance occurs:

$$alwaysIn(u_i, a_j, 0, 0), \qquad \forall i \in I, j \in J_M \setminus J_{M'}, rm_j = i \quad (24)$$

The constraint $alwaysIn(f, v, a, b)$ states that the value of a cumulative function f during the occurrence of an interval v must lay in the range $[a, b]$.

In the case of resource maintenance activities demanding only a partial unavailability, an auxiliary vector of cumulative functions **u'** is used to represent the use of the associated resources by a conflicting activity. Each of such functions is constrained to be 0 during the associated interval of partial unavailability:

$$u'_{j_m} = \sum_{j \in J'_{j_m} : c_{ij} = 1} pulse(m_{ij}, 1), \qquad \forall i \in I, j_m \in J_{M'}, rm_{j_m} = i \quad (25)$$

$$alwaysIn(u'_{j_m}, a_{j_m}, 0, 0), \qquad \forall j_m \in J_{M'} \quad (26)$$

The precedence between pairs of activities is directly stated with constraints involving the associated intervals of a. The first constraint below guarantees the chronological order between each of such pairs, and the second one that the presence of the latter interval depends on whether the former is also present:

$$end(a_{j_1}) + pd_{j_1 j_2} \leq start(a_{j_2}), \qquad \forall j_1 \in J_W, j_2 \in J_W, pc_{j_1 j_2} = 1 \quad (27)$$
$$presence(a_{j_2}) \rightarrow presence(a_{j_1}), \qquad \forall j_1 \in J_W, j_2 \in J_W, pc_{j_1 j_2} = 1 \quad (28)$$

The constraints to force the entire development of a well or its absence are defined between one of the first activities of each well and each of the remaining activities. Let $J_F \subseteq J_W$ be the set of the first development activities, i.e., those which are not preceded by other activities of the same well. Since it is possible that a well k has more than one activity in JF, let $J_{F1} \subseteq J_F$ be a set containing exactly one of such activities of each well. Each pair of activities from a well such that one belongs to J_{F1} and the other to $J_{NF1} = J_W \setminus J_{F1}$ are then both present or absent with the following constraint:

$$presence(a_{j_1}) = presence(a_{j_2}), \quad \forall j_1 \in J_{F1}, j_2 \in J_{NF1}, loc_{j_1} = loc_{j_2} \quad (29)$$

In order to limit the concurrency on wells and harbors, the cumulative functions of vector x are upper limited according to the type of each location:

$$x_k \leq 1, \qquad \forall k \in K_W \quad (30)$$
$$x_k \leq s_k, \qquad \forall k \in K_H \quad (31)$$

Similarly, the upper and lower limits of inventory on each pipelay vessel are imposed with constraints upon b and BH, respectively:

$$b_i \leq ic_i, \qquad \forall i \in I_V \quad (32)$$
$$bh_{ik} \geq 0, \qquad \forall i \in I_V, k \in K_H \quad (33)$$

In the case of loading activities, the inventory increase due to each interval is limited by its length. Thus, the following constraint limits the increase of inventory associated with each interval from definition (16):

$$heightAtEnd(bh_{ik}, m_{ij}) \leq ic_i * \frac{length(m_{ij})}{mal_i}, \qquad \forall i \in I_V, j \in J_H, k \in K_W, \; c_{ij} = 1, loc_j = k \quad (34)$$

The function $heighAtEnd(f, v)$ represents the variation of cumulative function f caused only by $stepAtEnd$ functions involving interval v.

The inventory is also constrained to be empty before a new load at a harbor:

$$alwaysIn(bh_{ik}, m_{ij}, 0, 0), \quad \forall i \in I_V, j \in J_H, k \in K_H, c_{ij} = 1, loc_j = k \quad (35)$$

With constraint (35), the shipment of each pipe p is always assigned to the last loading activity performed by a resource at the harbor of the pipe before the development activity of connection ca_p. In order to facilitate the satisfaction of (35), the implementation of (16) contains a variable negative step at the start of each load. Such step is aimed to perform small corrections in decisions regarding the weight loaded on previous activities, thus avoiding an excessive use of retraction during the search. Since that was only necessary for performance improvement on current solvers, we do not regard it as part of the model. For an explanation about the relation between search and the propagation of resource constraints, the interested reader is referred to the work of Laborie [13].

In order to reduce the search space associated with the number of loading activities, the bijection f can be used to define a symmetry breaking constraint. Without loss of generality, we can state that a well connection j_w is present if the associated loading activity j_h is also present:

$$presence(a_{j_h}) \rightarrow presence(a_{j_w}), \qquad \forall j_h \in J_H, j_w \in J_W, p \in P, \; f(p) = j_h, ca_p = j_w \quad (36)$$

In order to avoid the assignment of a pipe to a loading activity scheduled before its release date, it was necessary to define an additional vector $\mathbf{z}$. Each cumulative function z_i represents the maximum date of release of a pipe that is carried by resource i along time. One may observe that such cumulative function is actually representing a qualitative property, and that qualitative properties are usually modeled by means of state functions. The rationale for such design choice is that the syntax of cumulative functions is more appropriate in this case, since it enables to constrain that a property only changes at specific moments. Given that the variety of pipes that can be loaded is only altered when loading activities are performed, the change of the maximum date of release of pipes carried by a resource can only occur at the start of loading activities.

Each function z_i is composed in (37) of steps at the start of each loading activity performed by resource i. Its value is constrained by (38) to be always equal to the date of release of the pipe p associated by bijection f with the loading activity j that is performed, i.e., $z_i = rp_p$ for $f(p) = j$ when performing m_{ij}. Thus, the value of each function is always between 0 and **mard**, which is defined as the maximum date of release among all pipes. It is worth observing that the step due to each load is set to vary from $-mard$ to $+rp_p$ in order to account for the state of the resource right before the start of each load:

$$z_i = \sum_{j \in J_H, p \in P : c_{ij} = 1 \wedge f(p) = j} stepAtStart(m_{ij}, -mard, +rp_p), \qquad \forall i \in I_V \quad (37)$$

$$alwaysIn(z_i, m_{ij}, rp_p, rp_p), \qquad \forall i \in I_V, j \in J_H, p \in P, \; c_{ij} = 1, f(p) = j \quad (38)$$

Hence, it is possible to constrain each development activity of connection ca_p of a pipe p to be scheduled only when pipe p could have been previously loaded:

$$alwaysIn(z_i, m_{ij}, rp_p, mard), \qquad \forall i \in I_V, j \in J_W, p \in P, \; c_{ij} = 1, ca_p = j \quad (39)$$

It is worth of notice that constraint (39) forces that the load of a pipe p can only be made by an activity j associated by f with a pipe p' such that $rp_{p'} \geq rp_p$. It may appear that such constraint is more restrictive than necessary. However, theorem 1 guarantees that any solution can be represented with the model, and thus that (39) also serves as a symmetry breaking constraint for that reason.

Lemma 1. *Given an ORSP solution S that does not comply with constraint (39), there is always a solution S' equivalent to S with less violations to (39).*

Proof. The following operation can be applied to a solution S to obtain an equivalent solution S' such that the number of pipes of latest release date assigned to loading activities forbidden by (39) is reduced in at least one unit. Suppose that S is a solution that violates such constraint, and that p_1 is the pipe with the latest release date which is assigned to a loading activity j_2 such that $f(j_2) = p_2$ and $rp_{p_2} < rp_{p_1}$. Since there is one loading activity associated with each pipe from the same harbor hp_{p_1} with release date rp_{p_1} onwards and one of such pipes, p_1, is not assigned to neither of them, at least one of such activities, j_1, is either absent or assigned to load pipes with earlier release dates. Thus, there is an equivalent solution S', which differs from S by switching the assignment and schedule of activities j_1 and j_2, and therefore it has less pipes with the same release date as p_1 assigned to a loading activity forbidden by (39) than S has.

Theorem 1. *Constraint (39) does affect the correctness of the ORSP model.*

Proof. After a finite number of applications of the operation above, an equivalent solution satisfying (39) can be achieved from any ORSP solution.

Objective function The objective function represents the expected short-term production that would be accumulated from day 0 to day H with the schedule. It is depicted as the summation of the production rate triggered by the finish of each activity in the schedule multiplied by the time left until H:

$$\text{maximize} \qquad \sum_{j \in J_W} MAX(H - end(a_j), 0) * pr_j \tag{40}$$

Search phases The use of search phases is aimed to order the sets of variables of a problem in order to reduce the search effort. As a matter of fact, much of the work of ordering variables and values for assignment still remains to the solver in such a case. However, an outline of search phases according to the modeler point of view can help leveraging domain-specific knowledge to solve the problem.

In the ORSP model, we observed that it was better to assign first the intervals of a related to resource maintenance activities, starting with those demanding full unavailability. The rationale for such strategy is that such intervals are those with mandatory presence. Therefore, it is very likely that a delay in their assignments would cause more conflicts during the search. Besides, we relied on the premise that full unavailability periods are less sensitive to the assignment of activities to each resource, since they do not allow any concurrence.

4.2 Testing the ORSP model

The experimental evaluation conducted to test the model was based on data from a past scenario of the company. It comprises the activities to develop 171 wells using 73 resources. Such data was used to generate a number of instances by splitting the set of wells, and thus also splitting the corresponding sets of activities and pipes. For confidentiality and to facilitate comparisons, the value of the best solution found for each instance was used to define 100 in an arbitrary scale. In what follows, we will describe the instances and the experiments performed with them, present the results of the tests and discuss those results.

Experiment The model was tested using a set of instances first described in [21]. Instance O contains the entire set of activities of the past scenario used, which is partitioned approximately into halves for instances $H1$ and $H2$ as well as into quarters for instances $Q1$ to $Q4$. The same set of resources is considered in all cases. It is worth observing that those instances contain data regarding the details that were only introduced in the current work. Nevertheless, such data were ignored on the experiments of previous approaches. Table 1 summarizes how many activities, wells, pipes, rigs and vessels each instance has.

Table 1. Main characteristics of the tested instances

Instance	*Q1*	*Q2*	*Q3*	*Q4*	*H1*	*H2*	*O*
Activities	116	118	116	115	231	234	465
Wells	46	37	45	43	82	89	171
Pipes	17	17	13	19	32	34	66
Rigs	64						
Vessels	9						

The model was implemented using the OPL language and run using IBM Cplex Studio 12.2 with the CP Optimizer solver [11]. For each instance, it was run four times with different random seeds. The only modification to the standard solving parameters in such runs was that the cumulative function inference level was set as extended. The upper bound of each instance was set by running the MILP model M_F described in [22] on the Cplex solver with an empty schedule as starting solution. To achieve a tighter estimation, an additional constraint was defined to limit the start of each connection activity j, which is represented in M_F by variable S_j, according to the date of release of the pipes it must unload:

$$S_j \geq rp_p, \qquad \forall j \in J_W, p \in P, ca_p = j \quad (41)$$

The time limit of each run of either model was set as one hour in accordance to the end user expectation. The computer used had 4 Dual-Core AMD Opteron 8220 processors, 16 Gb of RAM and a Linux operating system.

Results The results of the runs are summarized in table 2. For each instance, it comprises the average (μ) and the standard deviation (σ) of the time to find the first solution, the production of such solution and the production of the last solution found according to a scale where the best solution found is 100. The upper bound corresponds to the optimal solution of the relaxed model M_F for instances $Q1$ to $Q4$ and $H1$, and to the upper limit reached at the solver halt for instances $H2$ and O. Since greater performance variations were observed only for instance O, the progress of each run to solve that instance is presented in figure 1. Such progress is depicted in terms of the best solution found along time.

Table 2. Summary of test results for each instance

	First solution found				Last solution		Upper bound	
	Time (s)		Value		Value		Value	Worst-case opt. gap
Instance	μ	σ	μ	σ	μ	σ		
Q1	0.44	0.02	99.1	0.3	100.0	0.0	100.1	0.1%
Q2	0.42	0.01	99.0	0.5	100.0	0.0	100.4	0.4%
Q3	0.60	0.02	99.4	0.2	100.0	0.0	100.1	0.1%
Q4	0.45	0.01	98.7	0.7	100.0	0.0	100.2	0.2%
H1	1.53	0.07	94.9	1.0	99.8	0.2	100.2	0.2%
H2	1.58	0.29	91.7	1.3	99.8	0.1	100.8	0.8%
O	4.57	0.24	83.2	1.5	98.9	1.5	107.2	7.2%

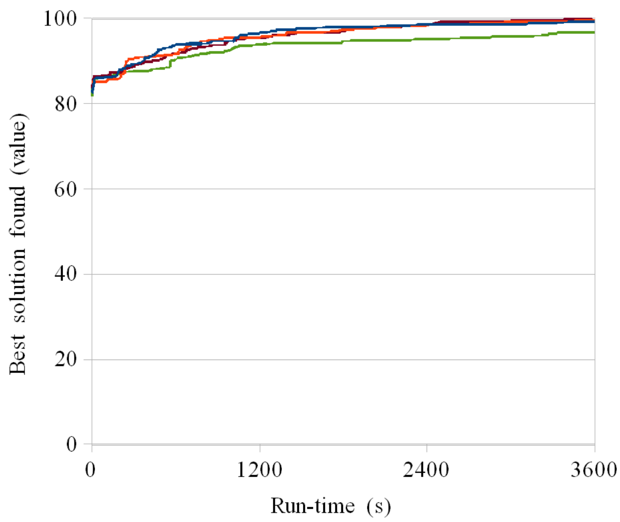

Fig. 1. Best solution found along time for independent runs on instance O

Discussion The results indicate that our approach was able to deliver good solutions within the time limit set by the end user. A worst-case optimality gap of less than 1% was achieved for all but the largest instance. For the scheduling of the whole set of activities, the gap was 7.2%. It is worth observing that about two thirds of the gap between the first solution and the upper limit vanished after one hour of search. In addition, figure 1 shows that there is a similar trend of improvement for most of the runs. Altogether, such observations evidence the system reliability to handle scenarios like those expected by the company.

Some scalability factors were observed as well. With a fixed time limit, the major impact noticed in solution quality is due to the number of activities that each instance has. The size of the instances also influenced the time and the quality of the first solution found. Among instances of similar size, the tightest gaps were achieved for the instances with a relatively small number of pipes, such as $Q3$ and $H1$. Therefore, the numbers of activities and pipes can be used for a preliminary assessment of instance hardness.

5 Benefits from CP

The greatest advantage sensed with the use of CP in this project was the easiness of prototyping. We were able to develop most of the project with only two professionals in full time dedication, who could deliver new and fully tested versions on a monthly basis. It was crucial that new models were available for evaluation as soon as possible for at least two reasons. First, many of the refinements asked each time by the clients of the project were attempts to describe their process. Second, it was often hard to gather an expressive amount of end users together to discuss the next steps of the development. In this context, the use of concise and easily maintainable models made it possible to work with a continuous deployment of new versions of the system. Hence, in spite of the number of rounds required to achieve a consensus of the problem definition, the short development cycle was essential to the conclusion and success of the project.

The end users showed a great interest in the results and in the technology beneath the system. That was caused by the positive prospect presented at the beginning of the project, which would facilitate a lot their daily work. Many of the questions that they raised referred to their expectation of what a good solution would be, and thus provided guidance to our work. In some of those occasions, they also wanted to understand how the solving process works. Since all of them were engineers and some had an IT background, the outline of the CP framework was easily assimilated. Nevertheless, there was a certain reluctance in accepting the use of a declarative paradigm if, when compared to alternatives like MP, CP was much more focused on feasibility than on optimality. Such opposition could be summarized by the idea that one should only give up the procedural control of a process if the incurred result was guaranteed to be the best possible one. However, the ability presented by the system to deal with large and overconstrained scheduling scenarios due to CP propagation mechanisms came to be considered important as well. In addition, some experiments like those reported in [23] helped to illustrate that it would not be worth to always assume certain assumptions regarding how the problem used to be solved manually. Those assumptions would require imposing unnecessary constraints, which were shown to harm considerably the long-term results on larger instances. With time, we were able to evaluate the solutions of the models through their rationale, and they were able to suggest modifications to the model at a more technical level.

6 Conclusion

This work approached the scheduling of an specialized fleet of oil rigs and pipelay vessels to develop offshore oil wells with the introduction of real-world constraints that have never been considered before. Two direct benefits have been observed by automating decisions related to the use of such resources. First, it reduces the burden over the professionals involved with the control of expensive and highly required machinery. If circumstances change, the time to perform a quick reschedule has passed from days to minutes. Second, a better return in terms of oil production can be pursued by generating and comparing many schedules. Experimental results indicate that a solution with worst-case optimality gap of 7.2% was achieved for a past scenario of the company. In addition, the anticipation of the development of a single well for an oil company usually would be enough to cover the expenses of an entire project like the present one. Hence, this article represents a detailed account on how to approach a type of problem that may interest many capital-intensive industries.

The difficulty to solve some optimization problems is often due to the lack of tools capable of leveraging the specificities of each application domain. We noticed that there is much to be gained by considering the use of Constraint Programming (CP) in such cases, for which reason the use of the technique is being considered in other projects within the company. As a consequence, this work has also focused on the rationale for using the technique as well as on the factors that led to the conclusion that other alternatives were not so appropriate. In a nutshell, CP allows tackling hard problems of large scale without giving up of a systematic approach, thus keeping the chance of finding an optimal solution. That is especially true for scheduling problems, which are quite common in many companies and are usually difficult to handle in such a way.

Acknowledgments The authors gratefully acknowledge Petrobras for authorizing the publication of the information here present. In addition, the opinions and concepts presented are the sole responsibility of the authors.

We also thank M.I. Harris, S. Harris, D. Ferber, A. Ciré and the anonymous reviewers for their valuable suggestions to improve the quality of this paper.

References

1. Accioly, R., Marcellino, F.J.M., Kobayashi, H.: Uma Aplicação da Programação por Restrições no Escalonamento de Atividades em Poços de Petróleo. In: Proceedings of the 34^{th} Brazilian Symposium on Operations Research, Rio de Janeiro, Brazil (2002)
2. Aloise, D.J., Aloise, D., Rocha, C.T.M., Ribeiro, C.C., Filho, J.C.R., Moura, L.S.S.: Scheduling Workover Rigs for Onshore Oil Production. Discrete Applied Mathematics, 154, pp. 695–702 (2006)
3. Baptiste, P., Laborie, P., Le Pape, C., Nuijten, W.: Constraint-Based Scheduling and Planning. In: Rossi, F., Beek, P. van, Walsh, T. (eds.); Handbook of Constraint Programming. Elsevier, Amsterdam, The Netherlands (2006)
4. Barták, R., Salido, M.: Constraint satisfaction for planning and scheduling problems. Constraints special issue, 16 (3) (2011)
5. BBC. Petrobras raises $70bn in record share offering. Available at: http://www.bbc.co.uk/news/business-11400277; accessed: March 13th, 2012 (2010)
6. British Petroleum: BP Statistical Review of World Energy, June 2011. Available at: http://bp.com/statisticalreview; accessed: March 13th, 2012 (2011)
7. Fick, J.: Petrobras Pumps First Crude from Massive Tupi Field Offshore Brazil. Rigzone. Available at: http://www.rigzone.com/news/article.asp?a_id=75679; accessed: March 13th, 2012 (2009)
8. Glinz, I., Berumen, L.: Optimization Model for an Oil Well Drilling Program: Mexico Case. Oil and Gas Business, 1 (2009)
9. Hasle, G., Haut, R., Johansen, B., Ølberg, T.: Well Activity Scheduling - An Application of Constraint Reasoning. In: Artificial Intelligence in the Petroleum Industry: Symbolic and Computational Applications II, pp. 209–228. Technip, Paris, France (1996)
10. Hooker, J.N.: Testing Heuristics: We Have It All Wrong. Journal of Heuristics, 1, pp. 33–42 (1995)
11. IBM: ILOG CPLEX Optimization Studio 12.2 documentation for ODM Enterprise. (2010)
12. Iyer, R., Grossmann, I.E., Vasantharajan, S., Cullick, A.S.: Optimal Planning and Scheduling of Offshore Oil Field Infrastructure Investment and Operations. Industrial & Engineering Chemistry Research, 37, pp. 1380–1397 (1998)
13. Laborie, P.: Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results. Artificial Intelligence, 143, pp. 151–188 (2003)
14. Laborie, P., Rogerie, J.: Reasoning with Conditional Time-Intervals. Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference, Coconut Grove, USA (2008)
15. Laborie, P., Rogerie, J., Shaw, P., Vilím, P.: Reasoning with Conditional Time-Intervals, Part II: An Algebraical Model for Resources. Proceedings of the 22st International Florida Artificial Intelligence Research Society Conference, Sanibel Island, USA (2009)
16. Lustig, I.J., Puget, J.-F.: Program Does Not Equal Program: Constraint Programming and Its Relationship to Mathematical Programming. Interfaces, 31, pp. 29–53 (2001)
17. Moura, A.V., Pereira, R.A., Souza, C.C. de: Scheduling Activities at Oil Wells with Resource Displacement. International Transactions in Operational Research, 25, pp. 659–683 (2008)

18. Nascimento, J.M. do: Ferramentas Computacionais Híbridas para a Otimização da Produção de Petróleo em Águas Profundas. Master thesis. Universidade Estadual de Campinas (Unicamp), Brazil (2002)
19. Pereira, R.A., Moura, A.V., Souza, C.C. de: Comparative Experiments with GRASP and Constraint Programming for the Oil Well Drilling Problem. In: Proceedings of the 4^{th} International Workshop on Experimental and Efficient Algorithms, pp. 328–340, Santorini Island, Greece (2005)
20. Petrobras: Plano de Negócios 2011–2015. Available at: http://www.petrobras.com.br/pt/quem-somos/estrategia-corporativa/downloads/pdf/plano-negocios.pdf; accessed: March 13th, 2012 (2011)
21. Serra, T., Nishioka, G., Marcellino, F.J.M.: A Constraint-Based Scheduling Of Offshore Well Development Activities with Inventory Management. In: Proceedings of the 43^{rd} Brazilian Symposium on Operations Research, Ubatuba, Brazil (2011)
22. Serra, T., Nishioka, G., Marcellino, F.J.M.: On Estimating the Return of Resource Aquisitions through Scheduling: An Evaluation of Continuous-Time MILP Models to Approach the Development of Offshore Oil Wells. In: Proceedings of the 6^{th} Scheduling and Planning Applications woRKshop (SPARK), ICAPS, Atibaia, Brazil (2012)
23. Serra, T., Nishioka, G., Marcellino, F.J.M.: Integrated Project Selection and Resource Scheduling of Offshore Oil Well Developments: An Evaluation of CP Models and Heuristic Assumptions. In: Proceedings of the Constraint Satisfaction Techniques for Planning and Scheduling Problems workshop (COPLAS), ICAPS, Atibaia, Brazil (2012)
24. Yergin, D.: The Prize: The Epic Quest for Oil, Money & Power. Free Press, New York, USA (2008)