Computers & Operations Research 39 (2012) 2944–2953

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Models and hybrid methods for the onshore wells maintenance problem

Christophe Duhamel [a,*], Andréa Cynthia Santos [b], Lucas Moreira Guedes [c]

[a] ICD-LOSI, STMR (UMR CNRS 6279), Université de Technologie de Troyes, France
[b] ICD-LOSI, Université de Technologie de Troyes, France
[c] Graduate Program in Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, MG, Brazil

ARTICLE INFO

Available online 16 March 2012

Keywords:
Workover rig
Scheduling
Vehicle routing
Column generation
Heuristics

ABSTRACT

Workover rigs are used in onshore basins but they are often in limited number and they may not attend all the maintenance requests. We consider here the problem of scheduling the rigs over a time horizon in order to minimize the total oil loss due to the idle production states. Three mixed integer linear models are proposed. The first one improves an existing scheduling-based formulation. The second one uses an open vehicle routing approach and the third one is an extended model for which a column generation strategy is developed. Several improvements are presented as well as two heuristics coupled with column generation. To our knowledge, the first optimal values for medium-size instances of the problem are presented in this paper. The results show the potential of the column generation and its interest in a practical context.

1. Introduction

Oil is currently the main source of world energy (oil 32.8%, coal 27.2% and gas 20.9% in 2009, see www.iea.org). It is also used for a large number of industrial products, such as plastics and pharmaceuticals drugs. The oil production, essentially by drilling and extracting, is a very hazardous activity which can have huge impacts on the environment. The growing demand leads to an intense extraction process, mostly done by using sophisticated and complex equipments such as pumpjacks for onshore wells. Thus, a regular maintenance is required in order to keep the system working. However, this does not prevent the occurrence of dysfunctions and the production has sometimes to be stopped. In this work, several mathematical formulations for the onshore wells maintenance problems are investigated.

Usually, the onshore wells maintenance is done using a workover rig, see Fig. 1. As this equipment is expensive, it is usually available in a limited number. Moreover, the requests following a failure spread out over the time. Consequently, there is often more requests to consider than the workover rigs can handle and one has to select which requests will be considered during the time horizon. The workover rigs can be different (heterogeneous) or similar (homogeneous). Since several types of maintenance have to be performed (cleaning, stimulation, reinstatement, etc.), service levels should be considered whenever these equipments are heterogeneous. Furthermore, the workover rigs are transported by dedicated trucks which are submitted to specific transit rules, according to the country. The vehicle fleet can also be homogeneous or heterogeneous.

We consider the problem of scheduling the workover rigs over the set of failing wells. It consists in both assigning the workover rigs to the wells and in finding optimal routes to perform the scheduled requests. For the sake of simplicity, this problem is referred here as the Workover Rig Problem (WRP). The objective is to minimize the total loss of petrol in a specified time horizon. Thus, deciding which workover rig performs a service (if so) on each well may take into account the well production, the rigs location, the road network and the service levels. Some strategies to solve this problem can be found in the literature: a scheduling-based formulation and a VNS heuristic are presented in [1]. The proposed heuristic is tested on realistic instances with up to 199 wells and with up to 11 workover rigs. The authors do not provide computational experiments on the model yet, since the suggested formulation does not seem to be the main focus. A GRASP with path relinking has been developed by [2]. Recently, a simulated annealing has been proposed on a variation of the WRP where the travel time is not considered [3]. Good results are reported on instances with up to 125 wells.

In this paper, we focus on mathematical formulations for the WRP. Providing optimal solutions for the WRP is an important issue from both an environmental and a financial point of view. The latest accidents on onshore and on offshore wells push forward the need for performing regular maintenance services. Moreover, according to [1], the traditional way of scheduling the workover rigs (without optimization techniques) in a medium-sized oil field in Brazil lead to oil

* Corresponding author.
E-mail addresses: christophe.duhamel@utt.fr (C. Duhamel), andrea.duhamel@utt.fr (A. Cynthia Santos), lucasguedes@cpdee.ufmg.br (L. Moreira Guedes).

0305-0548/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2012.02.026

Fig. 1. An example of a workover rig.

losses of about 2 millions dollars. Thus, there is room for substantial savings by applying operations research techniques.

The formulation suggested in [1] determines the absolute order each maintenance service is done. We first adapt it to allow wells to be left unattended which is a more realistic hypothesis. A second improvement consists in reducing the number of variables and constraints. Then, a lifting procedure is developed which significantly strengthens the formulation. Furthermore, a formulation combining the features of both the open vehicle routing problem (OVRP) and the team orienteering problem (TOP) is proposed. The idea is to focus on the sequence of wells attended by each workover rig rather than on the well order. Finally, we develop a model based on column generation together with some heuristics to compute sub-optimal solutions.

To our knowledge, this is the first work dedicated on designing and comparing both compact and extended formulations for the WRP. We have modeled the WRP by taking into account its main characteristics and sometimes in a more realistic way than in [1] (as by allowing unattended wells). The financial impacts and the fact that regions are practically managed by using clusters (dealing with medium size WRP problems) motivate the development and analysis on models and exact methods. Moreover, the time required to compute exact solutions (for some practical-size WRP problems belonging to clusters) is not significant compared to the horizon time planning. Finally, the models suggested in this work can also be used as a basis to develop hybrid methods.

The paper is organized as follows: basic notations, the problem definition and a bibliographical review are introduced in Section 2. The scheduling-based formulation and the proposed improvements are given in Section 3. Sections 4 and 5 are respectively devoted to the OVRP-based formulations and to the set covering-based formulation. Computational results are shown in Section 6 and final remarks are made in Section 7.

2. The WRP definition

The wells and the roads are represented by a digraph $G=(V,A)$, where V is the set of $n=|V|$ wells requiring a maintenance service and A is the set of arcs representing the shortest path between each pair of wells. The travel time t_{ij} is known for each arc $(i,j)\in A$. The daily oil production p_i and the maintenance service duration d_i are associated to each well $i\in V$. No depot is considered: once a service is done, the workover rig travels directly to the next well to perform a maintenance service. There are w workover rigs, each one being initially located at the last well on which it did a maintenance operation. Let e_{ij} be the travel time of rig i from its initial position to the well j.

In practice, the workover rigs and the vehicle fleet can be heterogeneous or homogeneous. The proposed formulations are designed to address the homogeneous case. The VRP and the scheduling based formulations can be easily adapted to work with a heterogeneous vehicle fleet. More modifications are required in the case of column generation because the columns must be indexed by route and vehicle type. Thus, a different subproblem is defined for each type of vehicle. Moreover, there is typically more requests than the workover rigs can handle. Thus some requests will be left unattended in the time horizon $[0; T]$, where T is the time by which all operations have to be done.

The WRP consists in minimizing the total oil loss due to production interruption while satisfying several constraints such as each attended request has to be addressed within a specified time horizon by at most one workover rig, and each workover rig has to be given a time-feasible schedule. It is worth mentioning that the objective is to minimize the total oil loss or, conversely, to maximize the total oil production. Thus, some wells might be left unattended even if they all can be visited in the time horizon. Such a situation is illustrated in Fig. 2 where three wells, A, B and C, have emitted a maintenance request. The travel time t_{ij} and the maintenance duration d_i are reported on the graph. The time horizon is $T=10$ and the production rates are $p_A=p_B=5$ and $p_C=20$. They all can be repaired by a rig initially located at node 0.

A first schedule (0, A, B, C) addresses all the maintenance requests and leads to a total oil production of $P_{0ABC}=35+20+20=75$. The second schedule (0, C, B) discards well A and it leads to a total oil production $P_{0CB}=100+10=110$. All the other feasible schedules lead to a smaller oil production. This happens since well C has the highest production rate. Performing its maintenance operation at the beginning of the time horizon leads to a large total oil production and it consumes half the available time. The remaining time is sufficient to attend well B and this leads to the highest possible total oil production. Thus (0, C, B) is optimal.

The WRP problem is closely related to some problems in the literature. For instance, the OVRP [4,5] consists in designing a set of routes for a fleet of vehicles in order to attend all customers demands, where the vehicles do not return to the depot. The vehicles travel from a starting point and end its tour at the last customers they visited. This feature is required in the WRP as well. However, there are some differences between the WRP and the OVRP. First, the vehicle capacity does not need to be taken into account in the WRP. Second, each client is not necessarily visited. Moreover, time is an important feature of the WRP since the oil production depends on the time by which the wells production can be reactivated.

The second problem related with the WRP is the TOP [6,7] which is a generalization of the orienteering problem (OP) [8]. The OP consists in finding a path for an agent from an origin to a destination. Each node is associated with a reward and the objective is to maximize the total collected rewards in a given

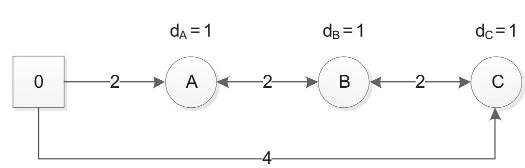

Fig. 2. Example where (0, C, B) dominates (0, A, B, C).

time horizon. In the TOP, the collection is done by a set of agents instead of a single one. Thus, a set of paths have to be found from an origin to a destination. The TOP differs from the WRP as the rewards are not time-dependent.

The time-dependent rewards for an OP are addressed in [9]. The work in [10] considers the problem of defining a schedule for a set of technicians within a given time horizon to perform maintenance services. Thus, each technician is assigned to a set of customers to be attended. The difference with the WRP is that each tour must begin and end at the depot.

Here, we use informations from both the OVRP, the TOP and the OP with time-dependent rewards to propose new formulations for the WRP. Some suggested improvements for the proposed formulations are inspired from [11].

3. Schedule-based formulation

The schedule-based model (S1) for the WRP consists in explicitly defining the order of maintenance of each well. Let $y_{ij}^k \in \{0,1\}$ be the decision variables determining whenever well j is the k-th attended by rig i. Moreover, variables $x_i \geq 0$ specify the time the maintenance begins at well i. This model is based on the assumption that rig i of type q_i can attend any well j of service level $l_j \leq q_i$. The schedule-based formulation presented in [1] is as follows:

$$\min \quad \sum_{j=1}^{n} p_j(x_j + d_j) \quad \text{s.t.} \tag{1}$$

$$\sum_{i=1}^{w}\sum_{k=1}^{n} y_{ij}^k \leq 1 \quad \forall j = 1 \dots n \tag{2}$$

$$\sum_{j=1}^{n} y_{ij}^k \leq 1 \quad \forall i = 1 \dots w, \ \forall k = 1 \dots n \tag{3}$$

$$\sum_{j=1}^{n} y_{ij}^{k+1} \leq \sum_{j=1}^{n} y_{ij}^k \quad \forall k = 1 \dots n-1, \ \forall i = 1 \dots w \tag{4}$$

$$l_j \sum_{k=1}^{n} y_{ij}^k \leq q_i \quad \forall i = 1 \dots w, \ \forall j = 1 \dots n \tag{5}$$

$$x_k \geq x_j + d_j + t_{jk} - M\left(2 - \sum_{h=1}^{s} y_{ij}^h - \sum_{h=s+1}^{n} y_{ik}^h\right) \quad \forall (j,k) \in A, \quad \forall s = 1 \dots n-1, \ \forall i = 1 \dots w \tag{6}$$

$$x_j \geq \sum_{i=1}^{w} e_{ij} y_{ij}^1 \quad \forall j = 1 \dots n \tag{7}$$

$$x_j \geq \left(1 - \sum_{i=1}^{w}\sum_{k=1}^{n} y_{ij}^k\right)(T - d_j) \quad \forall j = 1 \dots n \tag{8}$$

$$0 \leq x_j \leq T \quad \forall j = 1 \dots n \tag{9}$$

$$y_{ij}^k \in \{0,1\} \quad \forall i = 1 \dots w, \ \forall j = 1 \dots n, \ \forall k = 1 \dots n \tag{10}$$

The objective function (1) aims at minimizing the total oil loss over the time horizon. Eq. (2) guarantee that each stopped well is fixed by at most one workover rig. Constraints (3) state that each rig attends at most one well at each position in the sequence. Inequalities (4) specify that the wells are addressed consecutively by each rig. Inequalities (5) ensure a rig only attends compatible wells. Restrictions (6) determine the service start time which depends on the previous well in the sequence. Constraints (7) set the service start time of the first well in each schedule. Inequalities (8) do not belong to the original formulation in [1]. They ensure an unattended well cannot be operational before the end of the time horizon, for the objective function to remain consistent. Variables are defined in (9) and (10). This formulation contains $O(n^3 w)$ constraints and $O(n^2 w)$ decision variables.

3.1. Improving the S1 formulation

The first improvement we suggest to the S1 formulation (1)–(9) is to reduce the number of variables and constraints. As both the maintenance service level required by each well and the type of each rig are known beforehand, the set of workover rigs which can attend some kind of maintenance services is known. Thus, by only considering compatible assignments, many variables y can be eliminated and inequalities (5) are not necessary anymore.

Moreover, adjusting the M constant in constraints (6) can significantly strengthen the S1 formulation. We propose to set M to the time horizon T. Another improvement consists in lifting inequalities (7). Those constraints set a lower bound on the time well j is visited, whenever j is the first well of a schedule. Suppose that rig i starts its schedule by visiting well k then well j, see Fig. 3. Then, the time to start attending well j is at least the time to travel from the initial position to node k, to treat well k and to travel from k to j. This procedure is referred here as "two-hop-lift" and the corresponding new inequalities are given in (11).

The first part of inequalities (11) corresponds to the original lower bound given in constraints (7). The second sum defines the lower bound if well j is visited in the second position. The term α_i is defined in Eq. (12). It corresponds to the minimum time to attend well j whenever it is in the second position

$$x_j \geq \sum_{i=1}^{w} e_{ij} y_{ij}^1 + \sum_{i=1}^{w} \alpha_i y_{ij}^2 \quad \forall j = 1 \dots n \tag{11}$$

$$\alpha_i = \min_{k \neq j}\{e_{ik} + d_k + t_{kj}\} \quad \forall j = 1 \dots n \tag{12}$$

This idea can be extended to any positions in the tour. Thus, we also define the "three-hop-lift" in inequalities (13) and (14). However, the farther the position in the sequence, the higher the time to compute the constants (α_i, β_i and so on) and the smaller the benefit on the lower bounds. We noticed the use of "two-hop-lift" and "three-hop-lift" is a good trade-off between the quality of the lower bounds and the required computational effort

$$x_j \geq \sum_{i=1}^{w} e_{ij} y_{ij}^1 + \sum_{i=1}^{w} \alpha_i y_{ij}^2 + \sum_{i=1}^{w} \beta_i y_{ij}^3 \quad \forall j = 1 \dots n \tag{13}$$

$$\beta_i = \min_{k \neq j; l \neq j,k}\{e_{ik} + d_k + t_{kl} + d_l + t_{lj}\} \quad \forall j = 1 \dots n \tag{14}$$

Constraints (11) and (13) can be coupled with constraints (8) since the right hand sides are mutually exclusive, leading to the following aggregated constraints:

$$x_j \geq \sum_{i=1}^{w} e_{ij} y_{ij}^1 + \sum_{i=1}^{w} \alpha_i y_{ij}^2 + \left(1 - \sum_{i=1}^{w}\sum_{k=1}^{n} y_{ij}^k\right)(T - d_j) \quad \forall j = 1 \dots n \tag{15}$$

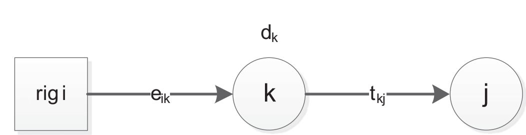

Fig. 3. Lifting for the lower bound constraints.

$$x_j \geq \sum_{i=1}^{w} e_{ij} y_{ij}^1 + \sum_{i=1}^{w} \alpha_i y_{ij}^2 + \sum_{i=1}^{w} \beta_i y_{ij}^3 + \left(1 - \sum_{i=1}^{w} \sum_{k=1}^{n} y_{ij}^k\right)(T - d_j) \quad \forall j = 1 \dots n \tag{16}$$

4. Formulation based on the open vehicle routing problem

The OVRP-based formulation (R) for the WRP relies on the sequence wells that are attended (relative order). Thus, variables y_{ij}^k specify whenever well j is treated just after well i using the workover rig k. Hence, the R formulation is more closely connected to the underlying transportation network. As before, variable x_i is the time the maintenance service starts at well i. Nodes 0 and $n+1$ denote respectively the artificial starting and the artificial ending nodes. The R formulation is as follows:

$$\min \quad \sum_{j=1}^{n} p_j (x_j + d_j) \quad \text{s.t.} \tag{17}$$

$$\sum_{j=1}^{n} y_{0j}^k = 1 \quad \forall k = 1 \dots w \tag{18}$$

$$\sum_{i=0}^{n} y_{il}^k - \sum_{j=1}^{n+1} y_{lj}^k = 0 \quad \forall l = 1 \dots n, \forall k = 1 \dots w \tag{19}$$

$$\sum_{i=0}^{n} y_{il}^k \leq 1 \quad \forall l = 1 \dots n, \forall k = 1 \dots w \tag{20}$$

$$\sum_{k=1}^{w} \sum_{i=0}^{n} y_{ij}^k \leq 1 \quad \forall j = 1 \dots n \tag{21}$$

$$x_j \geq \sum_{k=1}^{w} e_{kj} y_{0j}^k \quad \forall j = 1 \dots n+1 \tag{22}$$

$$x_i - x_j + (T + d_i + t_{ij}) \sum_{k=1}^{w} y_{ij}^k \leq T \quad \forall i = 1 \dots n, \forall j = 1 \dots n+1, j \neq i \tag{23}$$

$$x_j \geq \left(1 - \sum_{k=1}^{w} \sum_{l=1}^{n+1} y_{jl}^k\right)(T - d_j) \quad \forall j = 1 \dots n \tag{24}$$

$$0 \leq x_j \leq T \quad \forall j = 0 \dots n+1 \tag{25}$$

$$y_{ij}^k \in \{0,1\} \quad \forall i,j = 0 \dots n+1, \forall k = 1 \dots w \tag{26}$$

The objective function (17) aims at minimizing the total oil loss over the time horizon. For each workover rig, Eq. (18) specify each tour starts at well 0. Constraints (19) ensure the rig flow conservation at each node. Inequalities (20) ensure that each well is visited at most once. Constraints (21) determine that each node is treated by only one workover rig. Inequalities (22) and (23) determine the starting service time at each well. They are adapted from the classic Miller–Tucker–Zemlin constraints [12] and also ensure the subtour elimination. Constraints (24) state unattended wells cannot be operational before the end of the time horizon. Variables x_i and y_{ij}^k are respectively defined in (25) and (26). This formulation contains $O(n^3w)$ variables and $O(n^2)$ constraints.

We have improved this formulation by lifting inequalities (23) as in [11,13]. The lifted constraints are given in constraints (27)

$$x_i - x_j + (T + d_i + t_{ij}) \sum_{k=1}^{w} y_{ij}^k + (T - d_j - t_{ij}) \sum_{k=1}^{w} y_{ji}^k \leq T \quad \forall i = 1 \dots n, \forall j = 1 \dots n+1, j \neq i \tag{27}$$

Moreover, we have also developed generalized bounds which are given in constraints (28) and (29). Constraints (28) extend inequalities (22) by also taking into account the fact that node j can be visited after some node $i \neq 0$. In such a case, a lower bound is introduced. It corresponds to the time required to arrive at the previous node, to repair it, and to travel from it to node j

$$x_j \geq \sum_{k=1}^{w} e_{kj} y_{0j}^k + \sum_{k=1}^{w} \sum_{i=1}^{n} (e_{ki} + d_i + t_{ij}) y_{ij}^k \quad \forall j = 1 \dots n \tag{28}$$

The generalized upper bounds given in inequalities (29) consider the case when a well i is visited after well j. Thus, the time to treat wells j and i, and to travel from j to i is subtracted from the horizon time

$$x_j \leq M - d_j - \sum_{k=1}^{w} \sum_{i=1}^{n} (d_i + t_{ji}) y_{ji}^k \quad \forall j = 1 \dots n \tag{29}$$

5. Set-covering based formulation

The set-covering formulation (C) suggested for the WRP is an extended formulation resulting from a Dantzig–Wolfe decomposition of the previous model. Many VRPs have already been reformulated as set-covering problems, see for instance [7,14–17]. This formulation can be seen as selecting the best set of routes among all feasible routes. Since the number of feasible routes is potentially exponential, the set of routes cannot be explicitly enumerated. Thus, a specific strategy called column generation is applied to the linear relaxation. This column generation is then embedded within a branch-and-bound (branch-and-price) to find the optimal integer solution. In this section, we first present the basic version of a column generation strategy. Then we introduce several improvements (heuristics, a local search and a metaheuristic) to generate initial columns as well as improving columns.

Let $\Omega = \{r_1, \dots, r_{|\Omega|}\}$ be the set of feasible routes for a workover rig. A route r_k is an ordered set of visited wells. Let p_k be the oil production associated with route r_k and a_{ki} specify whether well i is visited by route r_k or not (the order needs not being specified). A decision variable $y_k \in \{0,1\}$ is associated with route r_k. It states whether r_k is selected or not. The C formulation, as opposed to the formulations suggested above, is set to maximize the total oil production (instead of minimizing the total oil loss). Both objectives functions are equivalent and the formulation is as follows:

$$\max \quad \sum_{k=1}^{|\Omega|} P_k y_k \quad \text{s.t.} \tag{30}$$

$$\sum_{k=1}^{|\Omega|} a_{kj} y_k \leq 1 \quad \forall j = 1 \dots n \tag{31}$$

$$\sum_{k=1}^{|\Omega|} y_k \leq w \tag{32}$$

$$y_k \in \{0,1\} \quad k = 1 \dots |\Omega| \tag{33}$$

The objective function (30) maximizes the total oil production using the selected routes. Constraints (31) ensure that each well is visited at most once. Restriction (32) states that at most w workover rigs are used. Variables y_k are defined in (33). This formulation contains an exponential number of variables and $O(n)$ constraints. It is known for having a better linear relaxation than the compact models.

As Ω cannot be explicitly handled, the column generation strategy works in the following way: the problem is first relaxed (linear master problem (LMP)) and at each iteration τ a subset $\Omega^\tau \subseteq \Omega$ is considered, giving the linear restricted master problem (LRMP). LRMP is then solved to optimality and dual variables are

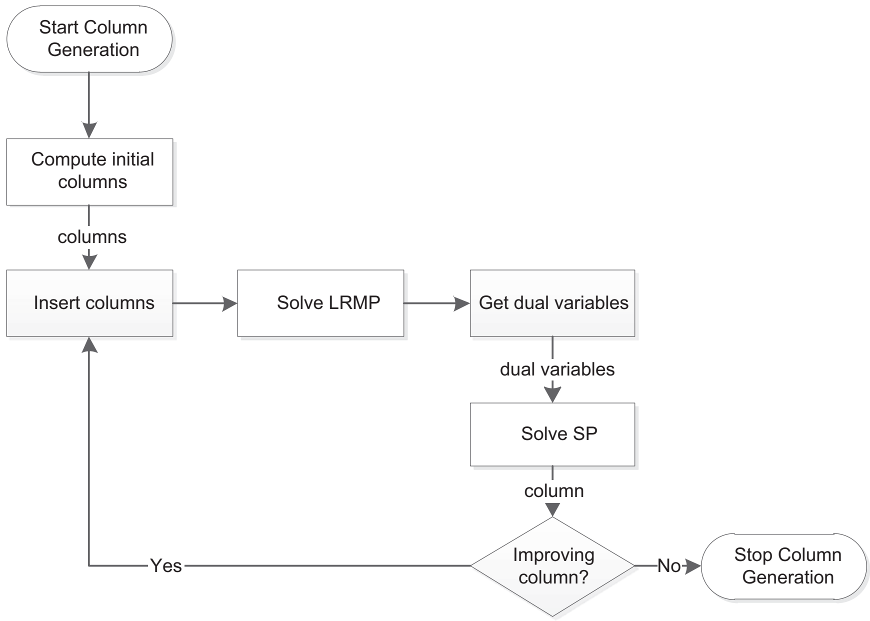

Fig. 4. The basic column generation strategy.

used to check if there exists some variable y_k, $k \in \Omega \backslash \Omega^\tau$ (i.e. route r_k) with a positive reduced cost. This is done by solving a subproblem (SP). If such a route r_k exists, the corresponding variable y_k is inserted, $\Omega^{\tau+1} = \Omega^\tau \cup \{k\}$ and the algorithm iterates. The procedure stops otherwise and the LRMP solution is also the optimal solution of LMP. Fig. 4 illustrates the basic column-generation algorithm.

5.1. The subproblem for the column generation strategy

At iteration τ, LRMP is solved to optimality and provides a pair of optimal primal and dual solutions. Let $\mu \geq 0$ be the dual variable associated to constraint (32) and let $\lambda_j \geq 0$ be the dual variable associated to restrictions (31) for well j. SP is used to identify a column (route) $y_k \in \Omega \backslash \Omega^\tau$ with a positive reduced cost $\overline{\mathrm{p}}_k$

$$\overline{\mathrm{p}}_k = p_k - \sum_{j=1}^{n} a_{kj}\lambda_j - \mu \geq 0 \quad (34)$$

SP corresponds to a variation of the elementary shortest path, namely the elementary shortest path problem with resource constraints (ESPPRC) and it maximizes (34). It can also be viewed as an orienteering problem with time-dependent rewards. If the computed route r_k is such that $\overline{\mathrm{p}}_k > 0$, the associated column y_k is an improving variable for LRMP. Otherwise no route can improve the current value of LRMP and its optimal solution is also optimal for LMP.

The main restrictions in SP are the total time and the number of visits at each well (at most one). The route travel time and the maintenance duration at the visited wells must fit within the time horizon T. Moreover, each well is visited at most once (hence the elementary path). For each well i, the gain s_i depends on its production. It is also modified by the dual variable λ_i, see Eq. (35)

$$s_i = (T - d_i - x_i)p_i - \lambda_i \quad (35)$$

The decision variable $y_{ij} \in \{0,1\}$ determines if well j is visited after well i and $z_i \in \{0,1\}$ states whether a well is visited or not. After reformulating the path cost (34) as a node-based function, the subproblem SP can be formulated as

$$\max \quad \sum_{i=1}^{n} ((T - d_i - x_i)p_i - \lambda_i z_i) - \mu \quad \text{s.t.} \quad (36)$$

$$\sum_{j=1}^{n} y_{0j} = \sum_{i=1}^{n} y_{i(n+1)} = 1 \quad (37)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ji} - \sum_{\substack{j=1 \\ j \neq i}}^{n} y_{ij} = 0 \quad \forall i = 1 \ldots n \quad (38)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^{n} y_{ji} = z_i \quad \forall i = 1 \ldots n \quad (39)$$

$$\sum_{i=1}^{n} \left(d_i z_i + \sum_{\substack{j=0 \\ j \neq i}}^{n} t_{ji} y_{ji} \right) \leq T \quad (40)$$

$$x_i - x_j + (T + d_i + t_{ij})y_{ij} + (T - d_j - t_{ji})y_{ji} \leq T \quad \forall i,j = 1 \ldots n,\ i \neq j \quad (41)$$

$$x_i + (T - d_i)z_i \geq (T - d_i) \quad \forall i = 1 \ldots n \quad (42)$$

$$x_j - t_{0j}y_{0j} - \sum_{\substack{i=0 \\ i \neq j}}^{n} (t_{0i} + d_i + t_{ij})y_{ij} \geq 0 \quad \forall j = 1 \ldots n \quad (43)$$

$$x_j + d_j + \sum_{\substack{i=0 \\ i \neq j}}^{n} (t_{ji} + d_i)y_{ji} \leq T \quad \forall j = 1 \ldots n \quad (44)$$

$$x_i \geq 0 \quad \forall i = 1 \ldots n \quad (45)$$

$$z_i \in \{0,1\} \quad \forall i = 1 \ldots n \quad (46)$$

$$y_{ij} \in \{0,1\} \quad \forall i,j = 0 \ldots n+1,\ i \neq j \quad (47)$$

The objective function (36) aims at maximizing the reduced cost. Constraints (37)–(39) ensure the elementary path. Constraint (40) sets the time limit. Inequalities (41) are the lifted Miller–Tucker–Zemlin constraints for the maintenance starting time. Constraint (42) ensures that an unvisited well does not produce any oil. Constraints (43) and (44) are, respectively, the lifted lower and upper bounds on the maintenance starting time. Variables are defined in (45)–(47). This formulation contains $O(n^2)$ constraints and $O(n^2)$ decision variables.

The bounds on the maintenance starting time (43) and (44) force an unvisited well to be unattractive, since its production will be null. However, as shown in Eq. (35), the interest of a well i also takes into account its dual variable λ_i. Thus, i becomes worthless in SP when its production drops below λ_i and the following upper bound can be defined:

$$x_i \le T - d_i - \frac{\lambda_i}{p_i} z_i \quad \forall i = 1 \ldots n \tag{48}$$

Property 1. *Constraint* (48) *is a valid cut.*

Proof. A node with a negative gain cannot belong to the optimal solution since removing it from the solution would lead to a better solution. Thus, forbidding negative gains for visited wells does not cut off the optimal solutions. □

As an example, consider a planning horizon $T=1000$ and a well i with a production rate $p_i=0.1$, with a maintenance service duration $d_i=100$ and whose dual variable is $\lambda_i=20$. It is visited immediately before well j such that $d_j=50$ and $t_{ij}=20$. The upper bound, as defined by constraint (44), is 830 with an associated gain $s_i=-14.8$. On the other hand, constraint (48) sets the upper bound at 700 and the associated gain is $s_i=0$.

SP is NP-hard, therefore only small to medium size instances can be efficiently solved by any MILP solver. A better practical approach is to use dynamic programming [18]. The label-correcting (LC) algorithm is an extension of the Bellman–Ford algorithm to handle shortest path problems with resource constraints (SPPRC). A label is associated to any partial path from node 0. Given a partial path p, its associated label $l=(c, h, f)$ stores its cost c, its resource consumption h and the label f used to create it. For the WRP, c refers to the oil production and h is the time spent so far. L_i is the list of labels corresponding to a partial path ending at node i. The label propagation mechanism from node i to node j consists in trying to extend all the labels in L_i to node j. In the basic version of the algorithm, a label can be propagated to j even if it has already been visited. To ensure the partial path to be elementary (and going from the SPPRC towards the ESPPRC), one solution consists in defining additional resources, one for each node. Such resource corresponds to the node availability when performing label propagation. Thus the label resource consumption h becomes a vector $(h_0, h_1 \ldots h_n)$ where h_0 is the time spent so far and $h_1 \ldots h_n$ are the nodes' availabilities when propagating. A label $l \in L_i$ such that $h_j=0$ corresponds to a partial path in which j has already been visited and l cannot be propagated to j. Additionally, a label for which the time consumption will exceed the time horizon T cannot be propagated as well. Otherwise, the label $l_1 \in L_i$ is propagated into label $l_2 \in L_j$ in the following way:

$$\begin{cases} l_2 \cdot h_0 = l_1 \cdot h_0 + t_{ij} + d_j \\ l_2 \cdot c = l_1 \cdot c + (T - l_2 \cdot h_0) \cdot p_j \\ l_2 \cdot f = l_1 \\ l_2 \cdot h_a = l_1 \cdot h_a & \forall a \ne j \\ l_2 \cdot h_j = 0 \end{cases}$$

The algorithm starts by creating the initial label $(0, 0, 1 \ldots 1, -1)$ at node 0 and then propagates it to the other nodes. Propagation is done as long as there is a node with unpropagated labels. Label dominance is used to reduce the number of labels and therefore to speed up the whole process. Given a node i, a label $l_1 = (c^1, h^1, f^1) \in L_i$ is said to dominate a label $l_2 = (c^2, h^2, f^2) \in L_i$ if it is associated to a higher oil production in a shorter time

$$l_1 \prec l_2 \Rightarrow (c^1 \ge c^2) \wedge (h^1 < h^2)$$

5.2. Improvements

Even if LC performs well, it still requires a lot of time for large instances. Thus, it is turned into a heuristic (HLC) by limiting the label propagation. Given a node i and a label $l \in L_i$, only the t best labels obtained after propagation are kept. When $t=n$, HLC corresponds to LC and when $t=1$, HLC corresponds to a greedy insertion heuristic. HLC is used to reduce the computational effort. Basically, the improvement consists in first calling HLC to identify improving columns. Upon success, LC needs not be called and the new columns are inserted into the LRMP for the next iteration. Otherwise, LC is called as in the basic version.

Another classic improvement consists in exploiting the fact that LC (and HLC) usually ends with a set of Pareto-optimal solutions. Thus one can provide LRMP with those solutions. Moreover HLC can be used to compute the initial LRMP columns. Even if LRMP could start with dummy columns, one can provide it with good initial columns. Thus, setting $\mu=0$, $\lambda=0$ and applying HLC leads to the computation of paths whose production is the highest. This might be troublesome since those paths might be very similar. Thus, HLC is used with a low k value (typically $k=2$) and it is called for every well set as first node in the path. The basic paths which consist in addressing a single repair request are inserted as well.

The next improvement results from a structural property of the problem: in the operational context, the processing times are much higher than the travel times. Repair times are usually measured in days while transportation times are computed in hours. Thus a good approximation consists in considering only the processing times and the problem becomes a scheduling problem. Let the attractiveness h_i of a well i be defined as the ratio $h_i = p_i / d_i$. Then the following property holds:

Property 2. *In the scheduling problem, the wells are visited in decreasing order of their attractiveness in any optimal solution.*

Proof. Consider wells i and j visited between wells k and l as in Fig. 5.

Let x_k be the starting maintenance time at node k. Thus x_i, x_j and x_l depend on the chosen subpath, either *kijl* or *kjil*. Let P_{ij} and P_{ji} respectively denote the total production when visiting i before j or after j. Since the travel time is not considered, x_l does not depend on the order and one can only consider the oil production of wells i and j. Then

$$\begin{cases} P_{ij} = (T - x_k - d_k - d_i) p_i + (T - x_k - d_k - d_i - d_j) p_j, \\ P_{ji} = (T - x_k - d_k - d_j) p_j + (T - x_k - d_k - d_j - d_i) p_i \end{cases}$$

In terms of total oil production, it is more interesting to visit i before j if

$$P_{ij} \ge P_{ji} \Leftrightarrow d_j p_i \ge d_i p_j \Leftrightarrow p_i / d_i \ge p_j / d_j \Leftrightarrow h_i \ge h_j$$

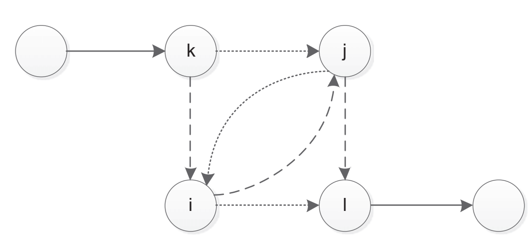

Fig. 5. Nodes i and j visited between nodes k and l.

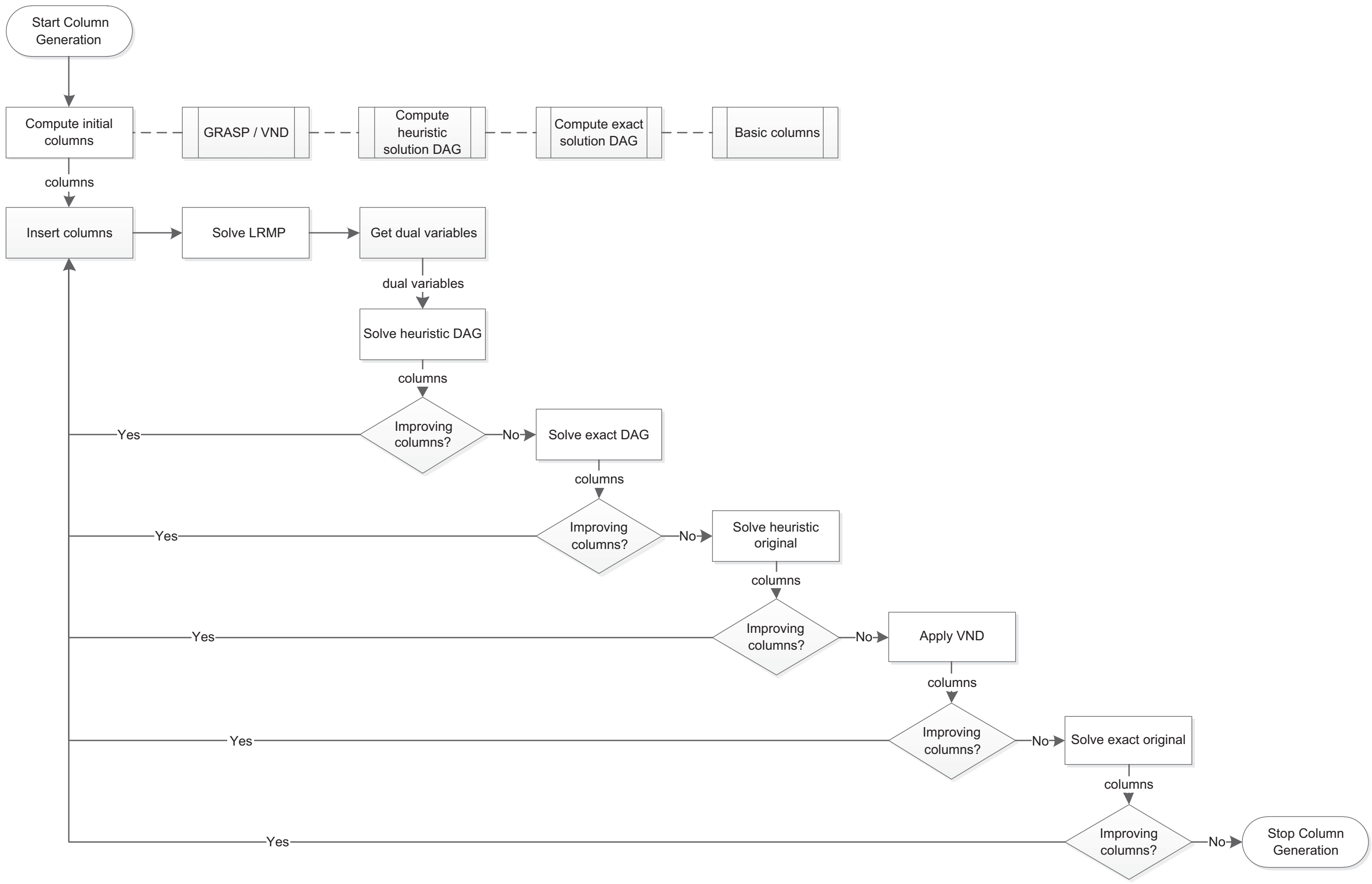

Fig. 6. The final column generation strategy.

Thus the attractiveness defines an order on the wells and this property must hold for any path in an optimal solution. □

When dropping the travel times, it is then sufficient to consider the graph obtained after keeping only the transportation arcs preserving the property. By definition, this is a directed acyclic graph (DAG), which simplifies the computation of improving columns. Let LC_DAG and HLC_DAG respectively denote the adaptation of LC and HLC to the DAG structure. Since the travel times are not taken into account in the computation of the production, this subset of columns might not contain some paths belonging to optimal solutions. Thus one still has to perform the computation on the original graph to meet the optimality criterion for the LRMP.

The last improvement consists in using a variable neighborhood descent (VND) as a local search to further limit the need to call LC. Three neighborhood structures $\mathcal{N}_k$, $k=1\dots3$ are defined. All use the best candidate policy. The VND strategy works as follows: let $\mathcal{N}_k$ be the active neighborhood structure at the current iteration and let p be the current solution. If an improving candidate $p' \in \mathcal{N}_k(p)$ is found, $k \leftarrow 1$ and $p \leftarrow p'$ for the next iteration. Otherwise $k \leftarrow k+1$ and the method iterates. The algorithm starts with $k=1$ and it stops when both three structures have failed to provide an improving candidate, i.e. $k=4$. The first neighborhood structure $\mathcal{N}_1$ considers all the swaps between two consecutive visited wells. The second one, $\mathcal{N}_2$, looks for all possible insertions of an unvisited well into p. The third one, $\mathcal{N}_3$, checks the exchanges of any unvisited wells with a visited one. The VND is applied to the 50 best paths computed by HLC. Moreover, it is also used as local search with slight modifications in a GRASP/VND metaheuristic to compute good initial solutions. In this context, the VND is given a fourth neighborhood structure, $\mathcal{N}_4$, which investigates all possible well exchanges between two routes. The GRASP constructive heuristic is a randomized version of the Clarke and Wright heuristic and 100 GRASP iterations are performed. The paths from the solutions obtained at the end of each GRASP iteration are then considered as initial columns for the LRMP and the final column generation strategy is shown in Fig. 6.

6. Results

The computational experiments were run on an Intel Core 2 Duo with 2.66 GHz clock and 4 GB of RAM memory. The compact formulations have been developed with OPL studio 6.3 and were tested using CPLEX 12 under default parameters. The column generation strategies have been developed in C++ with Concert. Comparison between the proposed formulations are measured in terms of time to prove optimality and time to solve the linear relaxation.

The instances have been generated using real characteristics of some medium size clusters from an oil basin in Brazil. The wells have been randomly chosen from the map of an oil basin in the Northeast region. Up to 60 wells are considered for a maintenance service which is chosen from a set of real maintenance requests (stimulation, cleaning, reinstatement for instance). In practice, the maintenance duration for requests of the same type can vary. We set an average duration (in days) for all the requests of the same type. The travel time typically requires a few hours. The wells production varies from 1 to 7 Bbl a day. The number of workover rigs varies from 2 to 5 and the time horizon is set to 15 days.

Table 1
Comparison of the schedule-based formulations.

w	n	O*	S1				S2				S3			
			cpu	Nodes	LR	% gap	cpu	Nodes	LR	% gap	cpu	Nodes	LR	% gap
2	3	26.72	0.02	23	22.71	15.00	0.00	0	24.41	8.66	0.01	0	25.65	4.00
	4	55.21	0.04	104	40.19	27.20	0.05	52	42.58	22.88	0.03	32	46.64	15.52
	5	57.40	0.22	474	37.65	34.41	0.17	264	40.32	29.76	0.12	135	44.39	22.67
3	4	44.22	0.05	71	40.19	9.11	0.03	6	41.40	6.38	0.02	0	43.15	2.42
	5	48.64	0.16	229	40.63	16.47	0.11	99	42.42	12.79	0.11	48	44.99	7.50
	6	59.63	1.77	2111	44.64	25.14	2.53	3011	46.83	21.47	0.47	280	50.17	15.86
	7	84.73	6.00	8290	60.79	28.25	4.99	3443	63.25	25.35	4.73	2320	67.17	20.72
	8	84.33	64.08	99 643	53.61	36.43	29.66	35 079	56.27	33.27	9.79	5883	60.80	27.90
	9	97.37	687.19	1 015 445	56.61	41.86	151.10	128 773	59.47	38.92	53.09	47 766	64.50	33.76
	10	125.41	236 027.52	5 044 223	76.80	38.60	2019.52	1 023 360	78.36	37.36	589.12	471 845	81.27	35.04
	11	157.95	–	–	–	–	155 567.55	5 676 657	93.41	40.86	56 949.38	1 763 093	96.43	38.95
	12	185.85	–	–	–	–	–	–	–	–	23 644.42	4 711 921	104.55	43.74
	13		–	–	–	–	–	–	–	–	–	–	–	–

Table 2
Comparison of the vehicle routing formulations.

w	n	O*	R1				R2			
			cpu	Nodes	LR	% gap	cpu	Nodes	LR	% gap
2	3	26.72	0.01	0	22.71	14.99	0.01	0	25.65	4.00
	4	55.21	0.02	26	40.19	27.20	0.03	0	47.57	13.83
	5	57.40	0.05	205	37.65	34.41	0.04	86	46.76	18.54
3	4	44.22	0.01	0	22.71	0.08	0.01	0	22.73	0.00
	5	48.64	0.02	0	40.19	9.11	0.01	0	43.15	2.43
	6	59.63	0.04	12	40.63	16.45	0.04	5	45.96	5.49
	7	84.73	0.12	422	44.64	25.13	0.06	132	52.90	11.28
	8	84.33	152.56	785 303	53.61	36.42	2.29	9824	70.03	16.95
	9	97.37	–	–	–	–	46.82	308 716	77.69	20.21
	10	125.41	–	–	–	–	76.92	441 942	104.74	16.48
	11	157.95	–	–	–	–	1105.55	5 155 053	127.58	19.22
	12	185.85	–	–	–	–	–	–	–	–
	13	–	–	–	–	–	–	–	–	–

Thus, the settings are consistent with the information provided in [1]. All the time units have been converted in seconds.

Results from the first experiment are reported in Table 1. Each line corresponds to an instance. S1 refers to the original schedule-based formulation (1)–(10). S2 corresponds to S1 coupled with the following strategies: two-hop-lifting (15), variables and constraints reduction, and setting $M = T$. S3 differs from S2 by the use of three-hop-lifting (16). Columns w and n respectively give the amount of workover rigs and of maintenance requests (wells). The optimal solution value is reported in column (O^*). For each formulation (S1, S2 and S3), the time in seconds to prove optimality (cpu), the number of nodes visited in the branch-and-bound tree (nodes), the value of the linear relation (LR) and the relative gap (% gap) are reported.

In spite of a big duality gap, the scheduling-based formulations are able to solve real instances with up to three rigs and 12 maintenance requests. Adding some improvements (lifting, reductions, adjustment) helps strengthening the initial formulation. The linear relaxation gaps have been improved on average by 3.58% for S2 and 8.7% for S3 when compared to S1. The computational time has also been significantly reduced: for example, the S1 takes about 3 days to solve the instance with three rigs and 10 maintenance requests, while S3 spends about 10 min.

Results for the OVRP-based formulation are given in Table 2. R1 refers to the formulation (17)–(26) with lifting constraints (27) and R2 corresponds to R1 with the generalized bounds (28) and (29). The gaps are smaller than those obtained by using the scheduling formulations. For instance, the gap averages are 11.67% using R2, against 22.34% using S3. However, the OVRP-based formulations solve instances with up to 11 wells only. We observe that the date and the horizon planning time (time window) significantly complicate the OVRP-based formulation. The generalized bounds strengthen the values associated to the variables which specify the starting service time. As a consequence, R2 is more efficient than R1. From our experience, the problem becomes clearly harder since some wells might be left

Table 3
Comparison of the column generation strategies.

w	n	O*	CG1			CG2		
			cpu	it.	% gap	cpu	it.	% gap
2	3	26.72	0.02	4	0.0	0.01	1	0.0
	4	55.21	0.02	5	0.0	0.01	1	0.0
	5	57.40	0.03	7	0.0	0.01	1	0.0
3	4	44.22	0.01	4	0.0	0.01	1	0.0
	5	48.64	0.03	6	0.0	0.01	1	0.0
	6	59.63	0.04	10	0.0	0.01	1	0.0
	7	84.73	0.04	10	0.0	0.01	1	0.0
	8	84.33	0.08	19	0.0	0.01	1	0.0
	9	97.37	0.10	21	0.0	0.01	1	0.0
	10	125.41	0.22	41	0.0	0.04	1	0.0
	11	157.95	0.12	44	0.0	0.11	2	0.0
	12	185.85	0.21	53	0.0	0.14	3	0.0
	13	220.23	0.49	64	0.0	0.17	4	0.0
	14	249.77	1.55	71	0.0	0.28	9	0.0

unattended. This results from the introduction of another combinatorial layer on whether a well is visited or not. Thus, the OVRP-based formulation shows a better performance when all nodes have to be visited.

Table 3 reports the results of the three column generation strategies on the same set of instances. *CG*1 refers to the basic column generation, using LC to generate the best improving column (Fig. 4). *CG*2 extends *CG*1 by first calling a GRASP/VND metaheuristic to compute initial columns, by computing improving columns on the DAG structure as well as performing a VND local search on the best solutions found by HLC, as shown in Fig. 6. For each strategy, the CPU time in seconds (cpu), the number of iterations (it.) and the relative gap to the optimal integer solution (% gap) are reported. With the exception of CG1 on the largest instance, all instances have been solved in less than 1 s. This is quite a strong reduction when comparing with the results from the compact models in Tables 1 and 2. CG2 improvements appear to speed up the process. The number of global iterations is reduced as well.

Since both variations produce results significantly better than the other models, one needs a set of larger instances to measure the differences. Table 4 displays the results on instances with up to 60 nodes and five rigs. The time limit has been set to 2 h and an asterisk (*) indicates that the optimal solution of the LRMP is fractional. The size of the largest instances (60 wells) already corresponds to realistic operational situations. One can note that the basic strategy is limited to 30–35 wells instances. The improvements in (CG2) allow to address instances with 60 wells.

Many optimal solutions of the LRMP are fractional in Table 4. In such cases a common heuristic (CG_H1) consists in using a solver (CPLEX) to compute the integer solution over the set of generated columns. This obviously provides an upper bound on the total oil loss. The quality of the solution can also be measured by the relative gap (relative difference between this upper bound and the lower bound given by the value of the fractional solution). We propose another column generation-based heuristic (CG_H2), which relies on the same principle except that only columns satisfying Property 2 are generated. Thus the column generation scheme is restricted to the LC_DAG, HLC_DAG column generators. The strategy is the same as CG_H1 when the resulting solution is fractional. Table 5 displays the results for both heuristics. For each instance and each heuristic, $\underline{O}$ is the value of the LRMP optimal solution, $\overline{O}$ is the value of the integer solution computed by the heuristic and *gap* is the relative gap in percents. Note that the gap for CG_H2 uses $\underline{O_2}$ instead of $\underline{O_1}$. The *cpu* columns report the total CPU time in seconds. The time required in the second step is nearly negligible as it never exceeds 0.5 s for CG_H1 and 0.9 s for CG_H2. CG_H1 always produces a solution at least as good as when using CG_H2. However, the time differs significantly for the largest instances. Thus, those two heuristics can already produce excellent solutions for real-life instances (with 50–60 wells). Moreover, given the quality of the solutions produced by CG_H2 and the time to compute them, CG_H2 may be a valuable choice for addressing even larger instances.

Table 4
Comparison of the strategies on larger instances.

w	*n*	*O**	CG1			CG2		
			cpu	it.	% gap	cpu	it.	% gap
3	15	529.54	2.09	76	0.0	0.06	3	0.0
	20	*722.81	67.44	86	0.0	0.59	3	0.0
	25	941.01	638.72	85	0.0	6.21	4	0.0
	30	1406.87	731.59	93	0.0	6.59	5	0.0
	35	*1837.24	3275.08	99	0.0	27.29	14	0.0
	40	2104.12	–	–	–	87.79	12	0.0
	45	*2558.00	–	–	–	312.60	8	0.0
	50	2684.48	–	–	–	459.38	18	0.0
	55	*3135.63	–	–	–	4408.67	15	0.0
	60	*3283.22	–	–	–	1276.96	9	0.0
4	15	426.45	1.43	70	0.0	0.06	6	0.0
	20	*478.54	38.17	117	0.0	0.13	4	0.0
	25	*768.79	431.17	130	0.0	1.52	5	0.0
	30	*1212.59	574.29	98	0.0	4.52	4	0.0
	35	1615.32	–	–	–	21.00	4	0.0
	40	*1884.92	–	–	–	25.11	4	0.0
	45	*2312.90	–	–	–	379.38	8	0.0
	50	*2463.12	–	–	–	352.98	12	0.0
	55	*2873.23	–	–	–	1350.96	12	0.0
	60	3037.58	–	–	–	991.74	14	0.0
5	15	366.41	1.13	68	0.0	0.04	4	0.0
	20	490.32	30.25	117	0.0	0.08	3	0.0
	25	649.67	284.49	150	0.0	0.27	3	0.0
	30	1040.69	565.98	217	0.0	4.05	6	0.0
	35	1409.91	–	–	–	32.10	9	0.0
	40	*1680.66	–	–	–	35.91	12	0.0
	45	*2080.57	–	–	–	206.67	8	0.0
	50	*2256.05	–	–	–	271.17	10	0.0
	55	*2630.23	–	–	–	1371.20	11	0.0
	60	*2810.24	–	–	–	794.81	20	0.0

Table 5
CG-based heuristics on larger instances.

w	*n*	CG_H1				CG_H2			
		$\underline{O_1}$	$\overline{O_1}$	% gap	cpu	$\underline{O_2}$	$\overline{O_2}$	% gap	cpu
3	15	529.54	529.54	0.00	0.06	529.54	529.54	0.00	0.03
	20	722.81	722.82	0.00	0.65	722.81	722.82	0.00	0.13
	25	941.01	941.01	0.00	6.21	941.01	941.01	0.00	0.11
	30	1406.87	1406.87	0.00	6.59	1406.87	1406.87	0.00	0.13
	35	1837.24	1837.30	0.00	27.35	1837.30	1837.30	0.00	0.36
	40	2104.12	2104.12	0.00	87.79	2104.12	2104.12	0.00	0.49
	45	2558.00	2558.20	0.01	312.65	2558.25	2558.43	0.02	0.84
	50	2684.48	2684.48	0.00	459.38	2685.13	2685.13	0.02	0.84
	55	3135.63	3137.44	0.06	4408.80	3136.20	3137.44	0.06	2.43
	60	3283.22	3283.49	0.01	1277.00	3283.34	3283.65	0.01	1.72
4	15	426.45	426.45	0.00	0.06	426.45	426.45	0.00	0.06
	20	478.54	478.60	0.01	0.20	578.54	478.60	0.01	0.15
	25	768.79	769.02	0.03	1.81	768.79	769.02	0.03	0.38
	30	1212.59	1212.79	0.02	4.60	1212.59	1212.79	0.02	0.42
	35	1615.32	1615.32	0.00	21.00	1615.56	1615.56	0.01	0.27
	40	1884.92	1885.15	0.01	25.52	1884.92	1885.15	0.01	0.45
	45	2312.90	2313.34	0.02	379.56	2312.93	2313.57	0.03	0.52
	50	2463.12	2463.40	0.01	353.02	2463.26	2463.40	0.01	0.83
	55	2873.23	2873.59	0.01	1351.06	2873.31	2873.59	0.01	1.90
	60	3037.58	3037.58	0.00	991.74	3037.58	3037.58	0.00	1.22
5	15	366.41	366.41	0.00	0.04	366.41	366.41	0.00	0.03
	20	490.32	490.32	0.00	0.08	490.32	490.32	0.00	0.04
	25	649.67	649.67	0.00	0.27	649.67	649.67	0.00	0.10
	30	1040.69	1040.69	0.00	4.05	1040.69	1040.69	0.00	0.28
	35	1409.91	1409.91	0.00	32.10	1409.91	1409.91	0.00	0.36
	40	1680.66	1681.46	0.05	36.35	1680.84	1681.54	0.05	0.76
	45	2080.57	2080.57	0.00	206.77	2080.71	2080.89	0.02	1.15
	50	2256.05	2256.19	0.01	271.53	2256.05	2256.19	0.01	1.63
	55	2630.23	2630.42	0.01	1371.38	2630.62	2630.79	0.02	2.09
	60	2810.24	2810.28	0.00	795.08	2810.37	2810.44	0.01	1.77

7. Conclusions

The WRP corresponds to an operational context in the oil industry where environmental and financial impacts are critical. Thus, providing optimal solutions (and not only good solutions) is an important issue that can lead to thousands of dollars savings.

In this work, we investigated several formulations for the problem. The schedule-based formulation has been first presented in [1]. We have extended it to allow wells to remain unattended which is more realistic since rigs are expensive equipments usually available is small quantities. Several improvements have been provided as well, especially the ‘‘two-hop-lift’’ and the ‘‘three-hop-lift’’ to strengthen the formulation. The second formulation uses features from the OVRP and from the OP. It has been improved by lifting MTZ constraints and by providing generalized upper and lower bounds. The third formulation consists in reformulating the problem as a set-covering problem. It uses a columns generation strategy to compute the optimal solution of the linear relaxation. This approach is improved by using heuristics to find initial columns and new columns. Moreover, we take advantage of the structure of the objective function to first generate the column in a subspace. If it fails to find improving columns, the search is then performed in the original space.

The numerical results show that the proposed improvements strengthen the formulations. The schedule-based (S3) and the OVRP-based (V2) formulations have the same level of performance: the gaps from (V2) are lower, but (S3) can solve a instance with one more well. Thus, they both are limited to small instances. The extended formulation is far more efficient since it can solve in less than 1 s all the instances the previous formulations solve. Besides, the optimal solution of the LRMP for all those instances is integer. The improvements allow to handle larger instances: (CG1) is limited to 30 wells, while (CG2) can address 60 nodes. Since the optimal solution of LRMP is sometimes fractional, one should develop a branch and price to find the optimal integer solution. We have used instead two column generation-based heuristics. They consist in computing an integer solution out of the set of generated columns. The first heuristic is based on CG2 while the second one relies on a restricted version is thus much faster. The solution they compute is within 0.1% of the optimality.

Those approaches remain affordable in an operational context when considering the planning time horizon. Extensions of the WRP addressing the fleet heterogeneity and uncertainties on the wells repair time are currently under investigation.

References

[1] Aloise DJ, Aloise D, Rocha CTM, Ribeiro CC, Filho JR, Moura LSS. Scheduling workover rigs for onshore oil production. Discrete Applied Mathematics 2006;154:695–702.
[2] Pacheco AVF, Ribeiro GM, Mauri GR. A GRASP with path-relinking for the workover rig scheduling problem. International Journal of Natural Computing Research 2010;1:1–14.
[3] Ribeiro GM, Mauri GR, Lorena LAM. A simple and robust simulated annealing algorithm for scheduling workover rigs on onshore oil fields. Computers and Industrial Engineering 2011;60:519–26.
[4] Derigs U, Reuter K. A simple and efficient tabu search heuristic for solving the open vehicle routing problem. The Journal of the Operational Research Society 2009;60:1658–69.
[5] Letchford AN, Lysgaard J, Eglese RW. A branch-and-cut algorithm for the capacitated open vehicle routing problem. The Journal of the Operational Research Society 2007;58:1642–51.
[6] Chao I, Golden B, Wasil E. The team orienteering problem. European Journal of Operational Research 1996;88:464–74.
[7] Boussier S, Feillet D, Gendreau M. An exact algorithm for the team orienteering problems. 4OR: A Quarterly Journal of Operations Research 2007;5:211–30.
[8] Vansteenwegen P, Souffriau W, Oudheusden DV. The orienteering problem: a survey. European Journal of Operational Research 2011;209:1–10.
[9] Erkut E, Zahng J. The maximum collecting problem with time-dependent rewards. Naval Research Logistics 1996;43:749–63.
[10] Tang H, Miller-Hooks E, Tomastik R. Scheduling technicians for planned maintenance of geographically distributed equipment. Transportation Research Part E 2007;43:561–609.
[11] Santos AC, Duhamel C, Aloise DJ. Modeling the mobile oil recovery problem as a multiobjective vehicle routing problem. Book series communications in computer and information science, vol. 14; 2008. p. 283–93.
[12] Desrochers M, Laporte G. Improvements and extensions to the Miller–Tucker–Zemlin subtour elimination constraints. Operations Research Letters 1991;10:27–36.
[13] Santos AC, Lucena AP, Ribeiro CC. Solving diameter constrained minimum spanning trees in dense graphs. In: Lecture notes in computer science, vol. 3059; 2004. p. 463–72.
[14] Butt S, Ryan D. An optimal solution procedure for the multiple tour maximum collecting problem using column generation. Computers and Operations Research 1999;6:427–41.
[15] Chabrier A. Vehicle routing problem with elementary shortest path based column generation. Computers and Operations Research 2006;33:2972–90.
[16] Oppen J, Lokketangen A, Desrosiers J. Solving a rich vehicle routing and inventory problem using column generation. Computers and Operations Research 2010;37:1308–17.
[17] Toth P, Vigo D. The vehicle routing problem. SIAM monographs on discrete mathematics and applications. Philadelphia; 2002.
[18] Feillet D, Dejax P, Gendreau M, Gueguen C. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. Networks 2004;44:216–29.