

# Global/local optimization strategies combined for waterflooding problems

Leonardo Correia de Oliveira<sup>1</sup>  · Silvana M. B. Afonso<sup>1</sup> · Bernardo Horowitz<sup>1</sup>

Received: 17 December 2014 / Accepted: 17 November 2015 / Published online: 21 December 2015  
© The Brazilian Society of Mechanical Sciences and Engineering 2015

**Abstract** In oil Reservoir Engineering application, a problem of great interest is the dynamic optimization of waterflooding management. In this work, the net present value (NPV) is taken as the function to be maximized, in which the allocated rates of wells are considered as design variables. Alternatively, the switching times of the control cycles can also be considered as design variables. This assumption increases flexibility to the management. Despite this, the formulation of this problem leads to a highly nonlinear, multimodal objective function. Therefore, to conduct the management, a hybrid optimization strategy is proposed here considering surrogate models. The hybrid strategy combines different methods at two different stages, a global and local. In this sense, the global search is driven by the genetic algorithm (GA) and the local search is driven by the sequential approximation optimization (SAO) method. The proposed methodology was successful in identifying wells that should be late started or shut-in before the end of the concession period and in handling different kinds of production strategies. It was also verified that an increasing on operation flexibility results in NPV improvement. Cycle duration variables proved to be useful in decreasing the number of design variables while maintaining recovery efficiency.

**Keywords** Hybrid strategies · Reservoir engineering · Waterflooding optimization · Surrogate model

## List of symbols

<b>A</b>	Coefficient matrix of constraints
<b>b</b>	Right-hand side of constraints
$c_w$	Cost of produced water
$c_{wi}$	Cost of injected water
$f$	Objective function
$\hat{f}$	Kriging model (surrogate function for objective function)
$F$	Fitness function
$g$	Inequality constraints
$\hat{g}$	Kriging model (surrogate function for inequality constraints)
$h$	Equality constraints
$\hat{h}$	Kriging model (surrogate function for equality constraints)
$I$	Injection wells
$k$	Permeability
$n_t$	Number of control cycles
$n_w$	Number of wells
$p$	Pressure
$P$	Producer well
$q$	Well rates
$Q$	Maximum rate allowed for well set of the field
$r$	Oil price
$t$	Time step
$T$	Concession period
$v$	Violation of the constraints
$x$	Design variables
$Z$	Gaussian function

## Greek symbols

$\alpha$	Ration between maximum production and injection rates
$\beta$	Unknown constant (for Kriging model)

Technical Editor: Celso Kazuyuki Morooka.

✉ Leonardo Correia de Oliveira  
leonardo.oliveira@ufpe.br

Silvana M. B. Afonso  
smb@ufpe.br

Bernardo Horowitz  
horowitz@ufpe.br

<sup>1</sup> Civil Engineering Department, Federal University of Pernambuco, Recife, PE 50670-901, Brazil

$\gamma$	Penalty parameter
$\delta$	Over injection parameter
$\Delta$	Size of the trust region
$\Delta\tau$	Time size of control cycles
$\lambda$	Activation parameter
$\sigma^2$	Variance

## 1 Introduction

In Oil Reservoir Engineering applications one problem of great interest is the dynamic optimization of production scheduling, considering constraints at total field rate meaning that all wells share common injection and production units. The waterflooding optimal management problem, which is by far the most commonly used method to improve oil recovery, is studied here. The objective is to maximize the exploitation of the field using as controls the rates of injector and producer wells.

The application of different optimization techniques in the field of upstream in the oil industry began in the 1950s and has been developing ever since. There are three major lines of study within the context of optimization of oil production [1]. One highlights the operation of oil fields, related to equipment and production facilities and wells. Another is related to the development oil field, related to quantity and distribution of development of wells (not exploration). The last one is related to the management oil field through the dynamic allocation of the flows of the wells.

There is a vast literature on the dynamic rate allocation optimization for waterflooding. One may classify the methods according to the degree of intrusion into the simulator code. The highly intrusive methods make use of the adjoint technique to compute the gradient of the objective function [2], and are among the most efficient methods [3–6].

Adjoint methods require a large programming effort to be implemented and are not available in all commercial simulator codes at the present time. The semi-intrusive methods make use of reduced order models [7, 8], or time of flight concept in streamline simulators to equalize water breakthrough in groups of producer wells [9]. Finally the non-intrusive methods use the simulator as a black-box and are purely data driven.

Algorithms typically use evolutionary techniques [10–12], pattern search methods [13] and surrogate-based methods [14]. Another class of derivative-free algorithms use approximate gradients of the objective function based on stochastic methods [15] and ensemble methods [16], which may be corrected by additional finite difference computations [17] or be incorporated into a quadratic interpolation model [18].

Some researchers believe that the creation of hybrid methods can result in tools that succeed in the optimization

process. Hybridization is already common practice in many fields. A hybrid algorithm may arise from the combination of two (or more) different strategies to solve one task. In principle, each strategy can be used independently, but this would produce solutions concentrated only in some specific characteristics. If the combination is efficient, unlike the independent use, chances are for a better solution to be found more quickly.

Güygüler [19] proposed a hybrid optimization technique, based on the genetic algorithm (GA) with the polytope algorithm (simplex) and the Kriging technique. Hybridization of genetic algorithm with simplex introduced hill climbing into the stochastic search and also allowed the use of proxies created and calibrated iteratively throughout the run and adopting simple surrogate models in place of expensive numerical simulations.

The choices of available algorithms to assist in the optimization process are numerous; however, the type of algorithm to be considered often depends on the characteristics of the problem. The optimization performed with the hybrid strategy considered here in is implemented in two steps. In the first step, a global search algorithm is responsible for identifying the region where the global optimum may be located. In the second step, a local search algorithm refines the solution found on the previous step. To perform a global search, a genetic algorithm (GA) was considered [20], and to perform the local search a sequential approximate optimization (SAO) was used [21, 22]. Kriging-based surrogates are used in both stages.

Some considerations are taken to improve the performance of the GA. Some of them are related to constraints handling, because it is well known that evolutionary algorithms have difficulty to deal with it. Other one is related to the surrogate model where the search process is performed, since it has an interest in a good characterization of the behavior of real functions, in the entire domain, without wasting evaluations on points from unpromising regions.

The main idea is to combine techniques into a tool where the user does not need to provide many settings to achieve the optimal solution of the problems.

## 2 Multimodality issues

The presence of multimodality in functions in some problems is characteristics that hinder the work of mathematical programming algorithms; therefore, to find the optimal solution for such problems, such algorithms need to be applied to multiple starting points. Various attempts represent multiple executions where different solutions can be found with no guarantee that the solution is the global optimum of the problem. Considering this, we could say that mathematical programming algorithm would not be

the best alternative to solve problems in which one seeks a global optimal solution and the involved function present multimodality characteristics.

Because of the difficulties, an alternative would be to adopt more robust search methods to deal with multimodality. In general, classes of algorithms that can perform this task are the evolutionary algorithms. However, the computational cost for this is quite high; therefore, when considering practical engineering problems, the use of such schemes is not a feasible option to be considered.

In short, we can say that both approaches mentioned above have useful features to obtain optimal solutions; however, the negative characteristics of them, in some situations, can avoid their use. To overcome the difficulties, and explore the positive aspects of the algorithms, an option that comes up is a combination of methods to create hybrid strategies and perform the search process.

### 3 Waterflooding problem

#### 3.1 Definition

Waterflooding is the most commonly used method to improve oil recovery and maintain the reservoir at a proper pressure level. Optimization techniques can be applied to improve waterflooding sweep efficiency through the propagation of the water front. In this sense, one problem of great interest is the dynamic optimization of producing scheduling, leading to optimal rate allocation to the injectors and producers. In this problem, the net present value (NPV) is considered as the objective function and the constraints are related to the rates: total field, limit values and voidage-type replacement [23].

#### 3.2 Mathematical formulation

Mathematically the waterflooding problem can be formulated as follows:

$$\begin{aligned}
 \text{Maximize: } NPV = f(\mathbf{q}) &= \sum_{t=0}^T \left[ \frac{1}{(1+d)^t} \cdot F_t(\mathbf{q}_t) \right] \\
 \text{subject to: } \sum_{p \in P} q_{p,t} &\leq Q_{l,\max}, \quad t = 1, \dots, n_t \\
 \sum_{p \in I} q_{p,t} &\leq Q_{inj,\max}, \quad t = 1, \dots, n_t \\
 q_{p,t}^l &\leq q_{p,t} \leq q_{p,t}^u, \quad p = 1, \dots, n_w, \quad t = 1, \dots, n_t \\
 \sum_{p \in P} q_{p,t} &\leq \sum_{p \in I} \alpha \cdot q_{p,t} \leq \delta \cdot \sum_{p \in P} q_{p,t}, \quad t = 1, \dots, n_t
 \end{aligned} \tag{1}$$

where  $\mathbf{q} = [q_1^T q_2^T \dots q_{n_t}^T]$  is the vector of well rates for all control cycles;  $\mathbf{q}_t = [q_{1,t}, q_{2,t}, \dots, q_{n_w,t}]$  is the vector of well rates at control cycle  $t$ ;  $q_{p,t}$  is the liquid rate of well  $p$  at control cycle  $t$ ;  $n_t$  is the total number of control cycles; and  $n_w$  is the total number of wells. In the objective function  $d$  is the discount rate and  $\tau_t$  is the time at the end of the  $t$ th control cycle. The cash flow at control cycle  $t$ , which represents the oil revenue minus the cost of water injection and water production, is given by

$$F(\mathbf{q}_t) = \Delta\tau_t \left[ \sum_{p \in P} (r_o q_{p,t}^o - c_w q_{p,t}^w) - \sum_{p \in I} (c_{wi} q_{p,t}) \right], \tag{2}$$

where  $\Delta\tau_t$  is the time size of the  $t$ th control cycle;  $P$  and  $I$  are the sets of production and injection wells, respectively;  $q_{p,t}^o$  and  $q_{p,t}^w$  are the average oil and water rates at the  $p$ th production well at  $t$ th control cycle;  $r_o$  is the oil price;  $c_w$  and  $c_{wi}$  are the costs of producing and injecting water.  $Q_{l,\max}$  is the maximum allowed total production liquid rate and  $Q_{inj,\max}$  is the maximum allowed total injection rate of the field. Superscripts  $l$  and  $u$  denote, respectively, the lower and upper bounds of design variables. Superscripts  $o$  and  $w$  denote, respectively, oil and water phases.

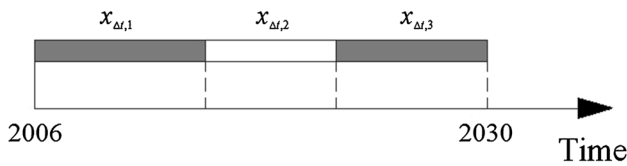
The last constraint requires that, for all cycles, the total injection rate belongs to an interval that goes from the total production rate to  $\delta$  times this value, where  $\delta \geq 1$  is the over injection parameter and  $\alpha$  is the ratio between  $Q_{inj,\max}$  and  $Q_{l,\max}$ . This is a more general form of the so-called voidage replacement constraint used by many researchers as a means to maintain the reservoir properly pressurized [3, 24–26]. Although reduction of the voidage replacement fraction, at the end of the production period may improve NPV [27], this practice may produce undesirable effects, such as drop pressure on the reservoir, consequently, this would result on the production of gas on it [28]. This constraint is not a mandatory part of the optimization process and may be deleted from the formulation if the user deems appropriate.

The design variables are the well rates in each control cycle. Let well rates be scaled by their respective maximum allowable field rates:

$$x_{p,t} = q_{p,t} / Q_{l,\max}, \quad p \in P; \quad x_{p,t} = q_{p,t} / Q_{inj,\max}, \quad p \in I. \tag{3}$$

Design variables,  $x_{p,t}$ , are then the allocated rate for well  $p$  at time at cycle  $t$ . We consider two alternative formulations for this problem: full capacity operation (FCO) and non-full capacity operation (NCO) [29].

The commonly used approach to these problems is to subdivide the concession period into a number of control cycles,  $n_t$ , whose switching times are fixed (see Fig. 1).



**Fig. 1** Control cycles’ switching times

Another step to make both of the mentioned strategies (FCO and NCO) more flexible is to consider the time of control switching as design variables. Although the number of design variables increases, the time of control switching can be made more intelligent, matching events like optimum water breakthrough, pressure needs of the reservoir and shutting-in of wells or the whole field. Also, due to this additional flexibility it is possible to obtain optimal management schedules with less control cycles.

Under previous paragraph considerations, the time interval of each control cycle, except for the last, is a design variable as depicted in Fig. 1. Scaling the time intervals by the concession period,  $T$ , one gets

$$x_{\Delta\tau,t} = \Delta\tau_t/T, \quad t = 1, \dots, n_t - 1. \tag{4}$$

Henceforth, we will call the previously defined fixed time strategies as FCO-FT and NCO-FT and the strategies which considers the time of control switching as design variables as FCO-VT and NCO-VT.

A general formulation for the waterflooding problem, considering all variables scaled, can be formulated as follows:

$$\begin{aligned} \text{Maximize: NPV} &= f(x) = \sum_{t=0}^T \left[ \frac{1}{(1+d)^t} \cdot F_t(x_t) \right] \\ \text{subject to:} & \sum_{p \in P} x_{p,t} \leq 1, \quad t = 1 \dots n_t \\ & \sum_{p \in I} x_{p,t} \leq 1, \quad t = 1 \dots n_t \\ & \sum_{t=1}^{n_t-1} x_{\Delta\tau,t} \leq 1 \\ & x_{p,t}^l \leq x_{p,t} \leq x_{p,t}^u, \quad p = 1 \dots n_w, \quad t = 1 \dots n_t \\ & x_{\Delta\tau,t}^l \leq x_{\Delta\tau,t} \leq x_{\Delta\tau,t}^u, \quad t = 1 \dots n_t - 1, \\ & \sum_{p \in P} x_{p,t} \leq \sum_{p \in I} \alpha \cdot x_{p,t} \leq \delta \cdot \sum_{p \in P} x_{p,t}, \quad t = 1 \dots n_t \end{aligned} \tag{5}$$

Computation of the objective functions of the above problems requires a complete reservoir simulation. As simulations can take several hours to run, the optimization algorithm is coupled to surrogate models constructed as described in the following sections.

## 4 Optimization strategies

### 4.1 Approximations: surrogate models

The surrogate functions have been used aiming a fast computation, and also to get a smooth and/or simplified function for the computationally expensive models (high fidelity). The essential is that high-fidelity functions (costly) can be replaced by simpler functions, which are used during the optimization process.

The Kriging data fitting scheme is the approach used in this work to build the surrogate models. The main idea of this model is to assume that errors are not independent but rather assume that they are correlated to the distance between corresponding points modeled by a Gaussian process around each sample point. The main advantages of this scheme are to easily accommodate irregularly distributed sample data, and the ability to model multimodal functions with many peaks and valleys. Moreover, Kriging models provide exact interpolation at the sample points [23].

Kriging technique involves interpolation or numerical regression (polynomial) of a set of results generated from the real model [30, 31]. Therefore, to build such a model, the first step is to generate a set of points (sample) that are unique and limited by the boundaries of the design space. This technique is known in the literature as design of experiments (DOE) [30, 31]. The most common approaches are: Latin Hypercube (LHS), Quasi-Monte Carlo (QMC), and Latinized Centroidal Voronoi Tessellation (LCVT) [32]. It has been found that for the specific problem of optimized waterflooding management LCVT outperforms the others in general and is, therefore, used in this study [33].

Once  $m$  sample points are generated, predictor expressions are developed to evaluate the function at untried design points. This is based on the “ordinary Kriging” approach which models the true function as [31]:

$$f(x) = \beta + Z(x) \tag{6}$$

where  $\beta$  is an unknown constant and  $Z(x)$  is a Gaussian random function with zero mean and variance  $\sigma^2$ .

After the construction of the approximation form, some model assessment strategies are required to check if a generated model is adequate. Such measure also provides some guidelines for selecting the best approximation model. The strategies considered are the same detailed in [23].

### 4.2 Hybrid strategy

#### 4.2.1 Global search: genetic algorithm (GA)

As previously mentioned, a GA, implementation from MATLAB, is the global strategy considered on optimization process. GAs, as powerful and broadly applicable

stochastic search and optimization techniques, are perhaps the most widely known types of evolutionary computation methods today. In the past decade, the GA community has turned much of attention to optimization problems in industrial engineering resulting in a large field of research and applications [34].

In general, a genetic algorithm has five basic components, as summarized by Michalewicz [35]:

- A genetic representation of solutions to the problem;
- A way to create an initial population of solutions;
- An evolution function rating solutions in terms of their fitness;
- Genetic operators that alter the genetic composition of children during reproduction;
- Values for the parameters of GA.

In general, GAs follow the steps below:

1. Generation of initial population;
2. Fitness evaluation for all individuals in population;
3. Application of genetic operators to generate new population (selection, crossover and mutation);
4. Checking convergence; if not reached, return to step 2 and repeat the process.

It is well known that evolutionary algorithms have problems to deal with constrained optimization, and it is not different in GA. Some techniques have been developed in attempt to overcome this difficulty. In most applications of GAs to constrained optimization problems, the penalty function method has been used [36]. In this work is used an adaptive penalty method [37] which does not require any type of user-defined penalty parameter and uses information from the population. Moreover, to make the global search procedure more effective a filter scheme is proposed in which the level of feasibility in the initial population is assured.

**4.2.1.1 Constraints handling** Constraint handling methods used in classical optimization algorithms can be classified into two groups [36]: (i) generic methods that do not exploit the mathematical structure (whether linear or nonlinear) of the constraint, and (ii) specific methods that are only applicable to a special type of constraints. Generic methods, such as the penalty function method, the Lagrange multiplier method, and the complex search method [38, 39] are popular, because each one of them can be easily applied to any problem without much change in the algorithm. However, the performance of these methods in most cases is not satisfactory.

On other hand, specific methods for constraints handling, such as the cutting plane method, the reduced gradient method, and the gradient projection method [38, 39],

are applicable either to problems only having convex feasible regions or to problems having a few variables.

Since GAs are generic search methods, most applications of them to constraint optimization problems have used the penalty function. The penalty function approach involves a number of penalty parameters which must be set right in any problem to obtain feasible solutions. This dependency of GAs performance on penalty parameters has led researchers to develop sophisticated penalty function approaches such as multi-level penalty functions [40], dynamic penalty functions [41], and penalty functions involving temperature-based evolution for penalty parameters with repair operators [42]. All these approaches require extensive experimentation for setting up appropriate parameters needed to define the penalty function [36].

In special situations, closed genetic operators (in the sense that when applied to feasible parents they produce feasible offspring) can be designed if enough design space knowledge is available [43]. Special decoders [44]—that always generate feasible individuals from any given genotype—have been devised, but no applications considering implicit constraints have been published.

The GA from the MATLAB global optimization toolbox has some alternatives to deal with constrained problems, but in this work, two alternatives of constraints handling were considered because the available methods could not find results for the studied cases.

**Chromosome repairing procedure:** For this process, initially a randomly population is generated. After this, the level of violation related to the linear constraints defined in Eq. (5) is verified. From this stage two parameters ( $\lambda^l$ ,  $\lambda^u$ ) are created. The first one is called inferior activation parameter, which refers to the individuals that activate the constraints by lower bounds

$$A \cdot (\lambda^l x) = b^l \tag{7}$$

The second parameter is created in a similar way. It is called superior activation parameter, which refers to the individuals that activate the constraints by upper bounds

$$A \cdot (\lambda^u x) = b^u \tag{8}$$

Next, a randomly value  $\lambda$  in the interval  $[\lambda^l, \lambda^u]$  is picked. This  $\lambda$  is named feasibility parameter. These parameters are generated for each unfeasible individual in the initial population. The use of  $\lambda$  guarantees that the imposed constraints will not be violated, but it is not correct to affirm that the individual is completely in feasible region because some of its coordinates could be out of its limits.

After the application of the feasibility parameter, the coordinates of the individual are ranked to be corrected if necessary. The coordinates are ranked in a descending order. In case of any coordinate be out of the limits, a coordinate correction is made to send the individual to the feasible region.

In case of upper bound violation, the excess is relocated to the next feasible coordinate as the design variables rank goes on. When more than one coordinate violate the limits, the excess is summed to be redistributed. When the lower bound is violated, the amount necessary to bring it to the lower bound value is taken from the next feasible coordinates.

After an unfeasible individual be transformed into a feasible one, its coordinates are relocated in the chromosome (the original positions). To consider this process is crucial for creating the initial population. The GA implementation considered here presents difficulty to solve the waterflooding problem, when it was used without any feasible individual in the initial population. In various tests considered, the optimization process has been finished without any feasible solution found.

It is important to mention that the above procedure is used to create the initial population for GA and to create some feasible samples to build the global surrogate model used in the global search of the optimization process (more details in Sect. 4.2.1.2).

The chromosome repairing is a good choice to handling the linear constraints, although it cannot be used as the only method for this task, because it would interfere directly in the search process. For example, if there are violated children generated by the crossover operator, they would be genetically modified by the repairing process, and this would change the main characteristics of the GA, in a sense that every violated children generated would be subjected to a kind of driven mutation process. It would happen in the children generated by the mutation operator too.

Adaptive penalty method: The adaptive penalty method (APM) presented in [39] adaptively quantifies the penalty coefficients of each constraint using information from the population such as the average value of the objective function and the level of violation of each constraint. The fitness function is written as:

$$F(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } \mathbf{x} \text{ is feasible,} \\ \bar{f}(\mathbf{x}) + \sum_{j=1}^m \gamma_j v_j(\mathbf{x}) & \text{otherwise} \end{cases} \quad (9)$$

where

$$\bar{f}(\mathbf{x}) = \begin{cases} f(\mathbf{x}) & \text{if } f(\mathbf{x}) > \langle f(\mathbf{x}) \rangle, \\ \langle f(\mathbf{x}) \rangle & \text{otherwise} \end{cases} \quad (10)$$

where  $\langle f(\mathbf{x}) \rangle$  is the average of the objective function values in the current population and  $\gamma_j$  is the penalty parameter for the  $j$ th constraint, and  $m$  is the number of constraints. It is defined at each generation by:

$$\gamma_j = | \langle f(\mathbf{x}) \rangle | \frac{\langle v_l(\mathbf{x}) \rangle}{\sum_{l=1}^m [\langle v_l(\mathbf{x}) \rangle]^2} \quad (11)$$

and  $\langle v_l(\mathbf{x}) \rangle$  is the violation of the  $l$ th constraint averaged over the current population. The idea is that the values of the penalty coefficients should be distributed in a way that those constraints which are more difficult to be satisfied should have a relatively higher penalty coefficient. Details of the proposed definition can be seen in [43].

To make clearer, consider Fig. 2 in which both feasible and infeasible solutions are shown. Among the six infeasible solutions, the individuals #3, #4, #5 and #6 have their objective function values (represented by opened circles), less than the average objective function and, according to the proposed method (see Eq. 10), this leads to  $\bar{f}(\mathbf{x}) = \langle f(\mathbf{x}) \rangle$ . The solutions #1 and #2 have objective function values which are greater than the population objective function average. Thus, for this case  $\bar{f}(\mathbf{x}) = f(\mathbf{x})$ .

4.2.1.2 Adaptive surrogate model The definition of a surrogate model that can define the overall behavior of the functions involved in the problem is not trivial, since it has an interest in a good characterization of the behavior of real functions without wasting evaluations on points from unpromising regions. In addition, when considering constrained optimization, the requirements are increased because it is important to build surrogate models that can represent the behavior from the real functions in feasible region while representing unfeasible region reasonably, since, in general, the global solution is located within the limits of the feasible region.

An alternative is to create a “weak” surrogate model which is enriched as the search process is performed. A scheme of surrogate model with addition of points on waterflooding optimization problem was considered with the particle swarm algorithm in the master thesis presented by Souza [45]. A similar scheme is considered in this work.

Initially, two samples are created: the first one is the initial sample, which has a representative number of points which belongs to the feasible region (it was considered 60 %—just a little more than half of the samples); and the

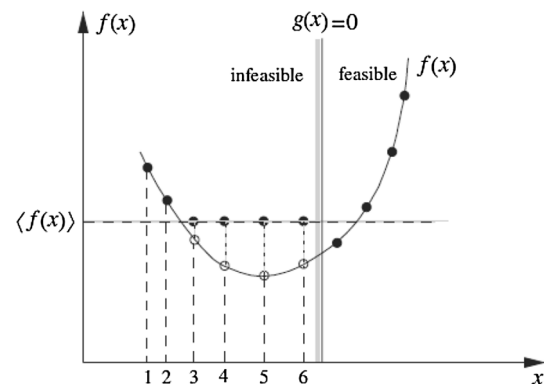


Fig. 2 A pictorial description of the function  $\bar{f}(\mathbf{x})$

second is the “update” sample, which has only feasible points to update the surrogate model (both samples are created with the chromosome repairing procedure help—previously mentioned).

The initial surrogate model is created with the initial sample; the optimization is performed with the GA on it. When the search is finished, the consistency of the solution is checked. If the accuracy of the model is not satisfactory, the model is updated with addition of new samples. The samples taken to update the model consist on the solution found by the GA and some points from the update sample (in this work where chosen three points randomly). The model is updated until there are points available on the update sample or the consistency of the solution is reached.

The consideration of the “solution” point to update the model improves the representation of the behavior on the region where it was found by GA; however, it should not be considered alone as it can be a local optimum. The addition of feasible samples improves model in other feasible regions.

The computational aspects of this strategy can be summarized as follows:

1. Create the initial sample (with 60 % in feasible region) and the update sample;
2. Compute the function evaluations of the initial sample;
3. Construct a global surrogate model for the functions;
4. Optimize the problem using GA with the approximated functions;
5. Evaluate the solution found by GA on the real function;
6. Check the global convergence criteria of the strategy:
  - 6.1. If reached, the process is finished;
  - 6.2. Otherwise, the solution found by GA is added to the initial sample as well as the points taken from the update sample (three points, which are evaluated on real functions), and returns to step 3.

#### 4.2.2 Local search: sequential approximate optimization (SAO)

A local strategy named Sequential Approximate Optimization (SAO) is used to compose the hybrid strategy in this study. The SAO methodology decomposes the original optimization problem into a sequence of optimization subproblems, confined to small subregions of optimization design space, called trust regions, whose extents are adaptively managed by the SAO strategy depending on surrogate accuracy [21, 46]. Surrogate functions (low cost) are created and used by the optimizer. Mathematically, each subproblem, at the  $k$ th SAO iteration, is defined as:

$$\begin{aligned}
 &\text{Maximize: } \hat{f}^k(x) \\
 &\text{subject to: } \hat{g}_i^k(x) \leq 0, \quad i = 1 \dots m \\
 &\quad \quad \quad \hat{h}_j^k(x) = 0, \quad j = 1 \dots n \\
 &\quad \quad \quad \mathbf{Ax} \leq \mathbf{b} \\
 &\quad \quad \quad \|x - x_c^k\|_\infty \leq \Delta^k \\
 &\quad \quad \quad x^l \leq x \leq x^u
 \end{aligned} \tag{12}$$

In the above equations,  $\hat{f}^k(x)$ ,  $\hat{g}_i^k(x)$  and  $\hat{h}_j^k(x)$  are, respectively, the surrogate objective and the constraints functions constructed for the current iteration,  $\mathbf{A}$  and  $\mathbf{b}$  are the matrix and right-hand side of linear constraints,  $x_c^k$  is the center point of the trust region,  $\Delta^k$  is the size of the trust region at the current iteration, and  $x^l$  and  $x^u$  are, respectively, the lower and upper bounds of the design variables. The initial trust region is a subregion taken from the entire domain which considers the initial point é central point.

Each subproblem formulated in Eq. (12) defines SAO iteration. The main steps involved in the computations are:

1. Compute the expensive and/or non-smooth objective function and constraints at the central point in the subregion;
2. Construct surrogate model in the subregion;
3. Optimize within the subregion using the surrogate objective function and constraints;
4. Compute the true objective function and constraints at the optimum identified in step 3;
5. Check for convergence;
6. Move/shrink/expand the subregion according to the accuracy of the approximated model compared to the true function and constraint values;
7. Impose local consistency;
8. Check for overall optimization convergence. If it is achieved stop the SAO procedure; otherwise return to step 3.

To update the trust region size for each optimization subproblem the approach considered takes into consideration the accuracy of surrogate functions against the true functions. For more details see [23]. More details about the SAO algorithm can be checked in [22, 23].

## 5 Examples

In this section, the tools described previously are employed in a study case. The studies are conducted using MATLAB. The reservoir numerical simulations are performed using IMEX Commercial Software from CMG [47].

### 5.1 Brush Canyon Outcrop Synthetic Case

This reservoir is composed of a complex geologic model constructed from geostatistical techniques and data from outcrop, which is close to a real reservoir model, although it is a synthetic case where the first studies were presented in [10].

The generated model reproduces typical of a depositional turbidity formations system in deep water, representing the most important type of reservoir found in sedimentary basins along the Brazilian coast. The petrophysical parameters were correlated, as can be seen in [48].

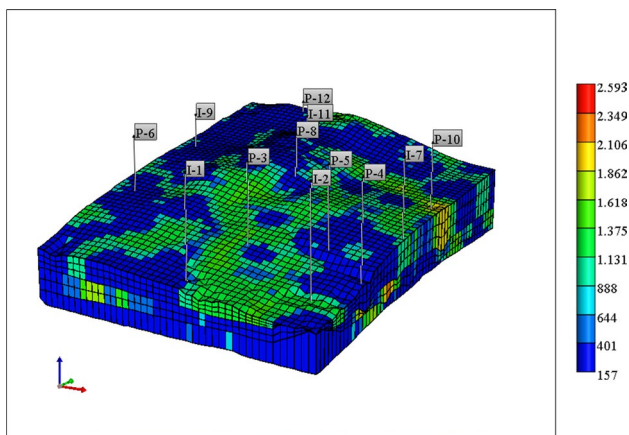
The features and the main characteristics of the model for this reservoir are presented in Table 1 and in Fig. 3.

Although the model incorporates a high degree of uncertainty, it is not considered in the studies performed here.

For the optimization problem, the production of the platform is limited to 5000 m<sup>3</sup>/day, and the total injected water is limited to 5750 m<sup>3</sup>/day. As the well rates are the design variables, the boundaries are imposed as the limitations on these rates.

**Table 1** Summary of model characteristics

Simulation mesh	43 (4,300 m) × 55 (5,500 m) × 6 (vary)
Porosity	16–28 %
Horizontal permeability ( $k_h$ )	157–2,593 mD
Vertical permeability ( $k_v$ )	30 % of $k_h$
Rock compressibility at 1,019 Kgf/cm <sup>2</sup>	$2 \times 10^{-7}$ (Kgf/cm <sup>2</sup> ) <sup>-1</sup>
Contact between fluids	Without contact WOC and GOC
Pressure of saturation ( $p_{sat}$ )	101.97 Kgf/cm <sup>2</sup>
Viscosity at $T_{res}$ and $p_{sat}$	0.77 cp
Gas–oil formation ratio (GOFR)	78.1 m <sup>3</sup> /m <sup>3</sup> std



**Fig. 3** Brush Canyon Outcrop model (permeability field)

For the producers, the rates can vary from 0 to 900 m<sup>3</sup>/day, and for the injectors, the rates can vary from 0 to 1500 m<sup>3</sup>/day. Normalizing the boundaries according the Eq. (3) gives to the producers a range of  $0 \leq x_{p,t} \leq 0.18$ ,  $t = 1 \dots n_p$ ,  $p \in P$ , while gives to the injectors a range of  $0 \leq x_{p,t} \leq 0.2609$ ,  $t = 1 \dots n_i$ ,  $p \in I$ .

The concession period is 24 years, and when considering the time of control switching as design variable, the related design variables are normalized according the Eq. (4).

A simplified version of VPL is used considering the revenues from the oil, and the production and injection costs (see Eqs. 1, 2). To calculate the NPV it is considered:  $r_o = 25.00/m^3$ ,  $c_w = 5.00/m^3$ ,  $c_{wi} = 2.00/m^3$  and 0.093 as internal return rate.

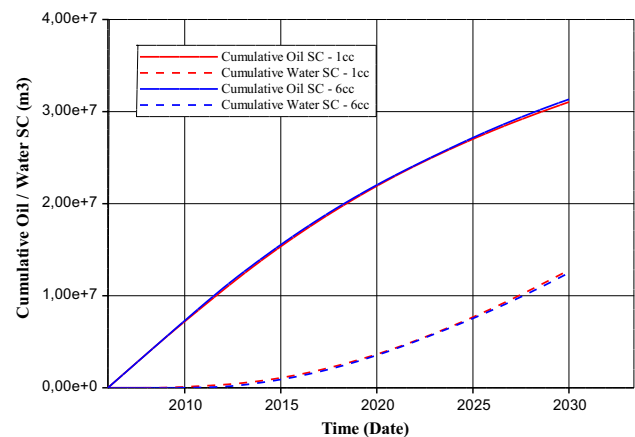
Next sections present solutions for two operation strategies: full capacity/fixed time (FCO-FT) and non-full capacity/variable time (NCO-VT). Normalized producer rates in the base case for all strategies are:  $x_{p,t} = 0.143$ ,  $t = 1 \dots n_p$ ,  $p \in P$ , while for all injectors,  $x_{p,t} = 0.2$ ,  $t = 1 \dots n_i$ ,  $p \in I$ . These values correspond to uniform rate allocation to all wells. The corresponding NPV value is  $\$2.6209 \times 10^8$ .

#### 5.1.1 Optimal solution: FCO-FT case

This section discusses solutions obtained for the fixed time, full capacity operation of the example. Two different numbers of control cycles were considered as shown in Fig. 4. In this case, the number of design variables is ten per control cycle, two less than the total number of wells.



**Fig. 4** Control cycles considered on concession period



**Fig. 5** Performance of FCO-FT with different control cycles



**Table 2** Optimization results for producer wells for six control cycles on FCO-FT case

Producers rates (m <sup>3</sup> /day)							
P-3	P-4	P-5	P-6	P-8	P-10	P-12	
500	0	900	900	900	900	900	
0	500	900	900	900	900	900	
0	500	900	900	900	900	900	
0	500	900	900	900	900	900	
0	500	900	900	900	900	900	
0	500	900	900	900	900	900	

**Table 3** Optimization results for injector wells for six control cycles on FCO-FT case

Injectors rates (m <sup>3</sup> /day)				
I-1	I-2	I-7	I-9	I-11
1466	0	1352	1466	1466
1466	668	684	1466	1466
1466	1139	213	1466	1466
1466	1352	0	1466	1466
1466	719	633	1466	1466
1466	1466	690	1466	662

The consideration of these two cases with different number of control cycles did not provide a big difference in the results obtained. The explanation for this can be perceived from Fig. 5.

Considering the hybrid strategy, the result found for the NPV, by the GA in the global search, was  $\$2.7173 \times 10^8$ , which does not represent great improvement when comparing with the base case.

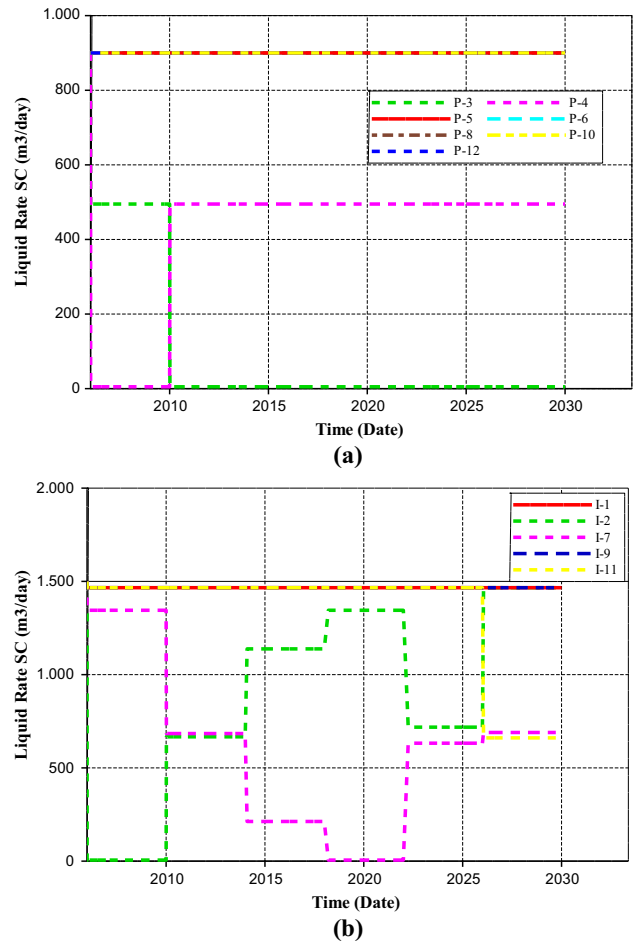
Taking the solution provided by the GA, and considering this solution as the initial point, for SAO in the local search, our best fixed time result for final NPV is  $\$3.1767 \times 10^8$ , which represents a 21.21 % of improvement over the base case.

For this case, a better result was already found. It is presented by Horowitz [23]. The NPV found in that work was  $\$3.2282 \times 10^8$ , which represents a 23.17 % of improvement over the base case (more details for this result can be found in [23]).

Tables 2 and 3 list optimum well rates for the case with six control cycles, which involves 60 independent design variables. As can be readily seen producer wells P-6, P-8, P-10, and P-12 are all operating at their maximum liquid rate. Figure. 6 shows the optimal rates of the remaining producers.

5.1.2 Optimal solution: NCO-VT case

Increasing the number of variables could afford an improvement on the NPV over the one on the base case,



**Fig. 6** Rate controls of selected well. **a** Producer wells and **b** injector wells

but, as can be seen in Fig. 5, it was considered six times more design variables and the result was not so different.

As mentioned in the mathematical formulation of the problem, a step that makes the production strategies more flexible is to consider the time of control switching as design variables. Although the number of design variables increases, the time of control switching can be made more intelligent. Due to this additional flexibility it is possible to obtain optimal management schedules with less control cycles.

**Table 4** Optimization results for producer wells for six control cycles on NCO-VT case

Time (days)	Producers rates (m <sup>3</sup> /day)						
	P-3	P-4	P-5	P-6	P-8	P-10	P-12
0	900	523	113	900	877	900	787
88	900	523	113	900	877	900	787
176	900	523	113	900	877	900	787

**Table 5** Optimization results for injector wells for six control cycles on NCO-VT case

Time (days)	Injectors rates (m <sup>3</sup> /day)				
	I-1	I-2	I-7	I-9	I-11
0	618	598	822	1500	1462
88	618	598	822	1500	1462
176	618	598	822	1500	1462

For this reason, this case considers the non-full capacity with variable time operation. Only the case with three control cycles is discussed. In this case, the range for the time of control switching variables is  $0 \leq x_{\Delta\tau,t} \leq 0.4$ ,  $t = 1 \dots n_t - 1$ , and the limits for the rates are the same presented in the FCO-FT case. The total number of design variables now is 38. The over injection parameter is taken as  $\delta = 1.15$  (see Eq. 1).

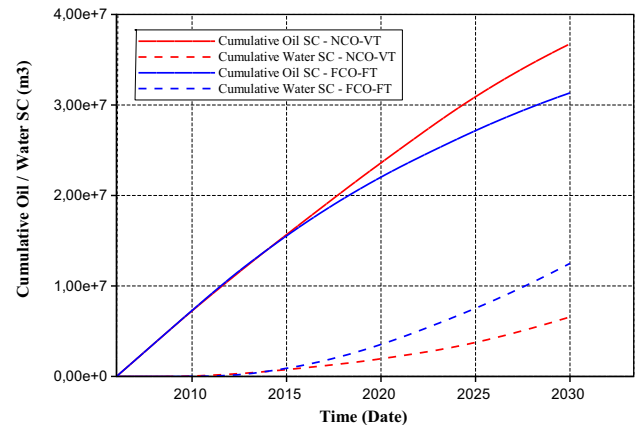
Again, the proposed methodology was considered. The result found for the NPV, by the GA in the global search, was  $\$3.5578 \times 10^8$ , and now, it represents a great improvement when comparing with the base case. But it is not the final result.

Considering this solution as the initial point, for SAO in the local search, our best result for final NPV is  $\$3.6233 \times 10^8$ , which is 14 % higher than the two previous solutions (FCO-FT cases), and corresponding to 38.25 % improvement when compared to the base case.

A result for this case can also be found in Horowitz [23]. The NPV found in that work was  $\$3.6100 \times 10^8$ , which represents a 37.74 % of improvement over the base case (more details for this result can be found in [23]).

Tables 4 and 5 show the values of the design variables, and the results obtained in both case can also be compared in Fig. 7.

The most flexible management strategy is NCO-VT, which creates more opportunities to increase NPV by the possibility of shutting-in and late starting of production and injection wells. The inclusion of control cycle duration in the formulation is effective in reducing significantly the number of design variables while maintaining recovery efficiency. On the other hand, it has been observed that this technique increases the nonlinearity of the problem which may become multimodal.

**Fig. 7** Performance of NCO-FT with different control cycles

## 6 Conclusions

The optimization of waterflooding management problem needs a large amount of function evaluation which means a large amount of reservoir simulation. It is computationally expensive. A powerful strategy to tackle this problem is the use of surrogate models based on data fitting techniques. Extensive comparative studies of performance in waterflooding problems indicate that LCVT sampling coupled with Kriging data fitting is the technique of choice to construct an effective surrogate model to deal with this kind of problems.

A hybrid global/local optimization strategy is used to solve the problem performing the optimization process in two steps, a global search (performed by GA) followed by a local search (performed by SAO). Although the SAO is an efficient optimization method, the hybrid methodology considered here overcomes the need to perform several initial point tries on reservoir optimization problems.

The waterflooding problem is formulated using liquid rates of wells as controllable variables. Duration of control cycles may also be included as design variables. The objective function is a simplified NPV. Different operational conditions may be modeled such as Full Capacity Operation (FCO) and Non-full Capacity Operation (NCO). In FCO, both injection and production units are forced operate at

their maximum capacity at all times whereas in NCO this condition is relaxed.

The most flexible management strategy is NCO with variable duration of control cycles. This creates more opportunities to increase NPV by the possibility of shutting-in and late starting of production and injection wells. The inclusion of control cycle duration in the formulation is effective in reducing significantly the number of design variables while maintaining recovery efficiency. On the other hand, it has been observed that this technique increases the nonlinearity of the problem which may become multimodal.

The cases considered involve a high nonlinear objective function and also presents several constraints functions. These aspects together turn out the problem too difficult to be solved by the GA (in charge of the first stage of the optimization). To improve its performance, some tools were considered to increase the capability to find a good feasible solution.

**Acknowledgments** The authors acknowledge the financial support for this research given by PRH-26, from ANP (National Petroleum Agency—Brazil), PETROBRAS, Foundations CMG, CNPq (National Research Council—Brazil), FACEPE (Support Foundation of Science and Technology of Pernambuco), and UFPE.

## References

- Horne RN (2002) Optimization application in oil and gas recovery. In: Pardalos PM, Resende MGC (eds) Handbook of applied optimization. Oxford University Press, New York
- Jansen JD (2011) Adjoint-based optimization of multiphase flow through porous media—a review. *Comput Fluids* 46(1):40–51
- Brouwer DR, Jansen JD (2004) “Dynamic optimization of waterflooding with smart wells using optimal control theory”, SPE 78278-PA. *SPE J.* 9(4):391–402
- Sarma P, Chen WH, Durlofsky LJ, Aziz K (2008) Production optimization with adjoint models under nonlinear control-state path inequality constraints. *SPE Reservoir Eval* 11(2):326–339
- Chen C, Wang Y, Li G, Reynolds AC (2010) Closed-loop reservoir management on the Brugge test case. *Comput Geosci* 14(4):691–703
- Chen C, Li G, Reynolds AC (2012) “Robust constrained optimization of short and long-term net present value for closed-loop reservoir management”. SPE 141314-PA. *SPE J.* 17(3):849–864
- Cardoso MA, Durlofsky LJ (2010) Linearized reduced-order models for subsurface flow simulation. *J Comput Phys* 229(3):681–700
- He J, Saetrom J, Durlofsky LJ (2011) Enhanced linearized reduced-order models for subsurface flow simulation. *J Comput Phys* 230(23):8313–8341
- Alhuthali AH, Datta-Gupta A, Yuen B, Fontanilla JP (2009) Field applications of waterflooding optimization via rate control with smart wells. In: SPE reservoir simulation symposium, SPE 118948, 18–20 February, The Woodlands, TX
- Oliveira DF (2006) Production optimization techniques for petroleum reservoirs: derivative free approaches to dynamic rate allocation for injection and production. Master thesis, Civil Engineering Department, UFPE, Recife, Brazil (in Portuguese)
- Almeida LF, Vellasco MMBR, Pacheco MAC (2010) Optimization system for valve control in intelligent wells under uncertainties. *J. Pet. Sci. Eng.* 73:129–140
- Souza SA, Afonso SMB, Horowitz B (2010) Optimal management of oil production using the Particle Swarm Algorithm and Adaptive Surrogate Models. In: XXXI Iberian Latin-American Congress on Computational Methods in Engineering, Buenos Aires, 15–18 November, Argentina
- Asadollahi M, Dadashpour M, Kleppe J, Naevald G (2009) Production optimization using derivative free methods applied to Brugge field. In: IRIS-Gubkin-StatOil-NTNU workshop in production optimization, Torondheim, Norway, 28th September
- Queipo NV, Goicochea JV, Pintos P (2002) Surrogate modeling-based optimization of SAGD process. *J. Pet. Sci. Eng.* 35:83–93
- Wang C, Li G, Reynolds AC (2009) “Production optimization in closed-loop reservoir management”, SPE 109805. *SPE J.* 14(3):506–523
- Chen Y, Oliver DS (2010) “Ensemble-based closed-loop optimization applied to Brugge field”. SPE 118926-PA. *SPE Res. Eval.* 13(1):56–71
- Xia Y, Reynolds AC (2013) An optimization algorithm based on combining finite difference approximations and stochastic gradients. In: SPE reservoir simulation symposium, SPE 163613, 18–20 February. The Woodlands, TX
- Zhao H, Chen C, Do S, Li G (2011) Maximization of a dynamic quadratic interpolation model for production optimization. In: SPE reservoir simulation symposium, SPE 141317, 21–23 February. The Woodlands, Texas, USA
- Güyagüler B (2002) Optimization of well placement and assessment of uncertainty. Doctor of philosophy thesis, Stanford University, Stanford, CA
- Goldberg DE (1989) Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Reading, MA
- Alexandrov N, Dennis JR, Lewisand RM, Torezon V (1997) A trust region framework for managing the use of approximation models in optimization. NASA/CR-201745, ICASE report no. 97-50
- Eldred MS, Giunta AA, Collis SS (2004) Second-order corrections for surrogate-based optimization with model hierarchies. Paper AIAA-2004-4457 in proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY
- Horowitz B, Afonso SM, Mendonça CV (2013) Surrogate based optimal waterflooding management. *J Pet Sci Eng* 112:206–219
- Naevald G, Brouwer DR, Jansen JD (2006) Waterflooding using closed-loop control. *Comput Geosci* 10:37–60
- Van Essen GM, Zandvliet MJ, Van den Hof PMJ, Bosgra OH, Jansen JD (2009) “Robust waterflooding optimization of multiple geological scenarios”, SPE 102913-PA. *SPE J.* 14(1):202–210
- Asadollahi M (2012) Waterflooding optimization for improved reservoir management. PhD thesis, NTNU, Norway
- Lorentzen RJ, Shafieirad A, Naevald G (2009) Closed loop reservoir management using the ensemble Kalman filter and sequential quadratic programming. SPE 119101. In: SPE Reservoir Simulation Symposium, 18–20 February, The Woodlands, TX
- Asadollahi M, Naevald G (2010) Selection of decision variables for large-scale production optimization applied to the Brugge field, SPE 136378. In: SPE Russian oil and gas conference and exhibition, 26–28 October, Moscow, Russia
- Horowitz B, Afonso SM, Mendonça CV (2010) Using cycle switching times as design variables in optimum waterflooding management. In: EngOpt—2nd international conference on engineering optimization, Instituto Técnico Superior de Lisboa, Lisbon, Portugal

30. Keane AJ, Nair PB (2005) Computational approaches for aerospace design: the pursuit of excellence. Wiley, Hoboken, 2005, p 602. ISBN 0470855401
31. Forrester A, Sobester A, Keane A (2008) Engineering design via surrogate modeling: a practical guide. Wiley, Chichester, p 228. ISBN 0470060689
32. Giunta AA, Wojtkiewicz SF, Eldred MS (2003) Overview of modern design of experiments methods for computational simulations. Paper AIAA-2003-0649, proceedings of the 41st AIAA aerospace sciences meeting and exhibit, Reno, NV
33. Afonso SMB, Horowitz B, Willmersdorf RB (2008) Comparative study of surrogate models for engineering problems. In: Proceedings of the 7th international ASMO-UK/ISSMO conference on engineering design optimization, Bath, UK
34. Gen M, Cheng R (2000) Genetic algorithm and engineering optimization. Wiley, New York
35. Michalewicz Z (1996) Genetic Algorithm + Data Structure = Evolution Programs, 3rd edn. Springer-Verlag, New York
36. Deb K (2000) An efficient constraints handling method for genetic algorithms. *Comput Methods Appl Mech Eng* 186(2–4):311–338
37. Lemonge ACC, Barbosa HJC, Borgesc CCH, Silva FBS (2010) Constrained optimization problems in mechanical engineering design using a real-coded steady-state genetic algorithm. In IX Congreso Argentino de Mecánica Computacional/II Congreso Sudamericano de Mecánica Computacional/XXXI CILAMCE—Congreso Ibero-Latino-Americano de Métodos Computacionales en la Ingeniería
38. Deb K (1995) Optimization for Engineering Design: Algorithms and Examples. Prentice-Hall, New Delhi
39. Reklaitis GV, Ravindran A, Ragsdell KM (1983) Engineering Optimization Methods and Applications. Wiley, New York
40. Homaifar A, Lai SH-V, Qi X (1994) Constrained optimization via Genetic Algorithms. *Simulation* 62(4):242–254
41. Joines JA, Houck CR (1994) On the use of nonstationary penalty functions to solve nonlinear constrained optimization problems with gas. In: Michalewicz Z (ed) Proceedings of the international conference on evolutionary computation. IEEE Press, Piscataway, pp 579–584
42. Michalewicz Z, Attia N (1994) Evolutionary optimization of constrained problems. In: Sebald AV, Fogel LJ (eds) Proceedings of the third annual conference on evolutionary programming. World Scientific, Singapore, pp 98–108
43. Schoenauer M, Michalewicz Z (1996) Evolutionary computation at the edge of feasibility. In: Voigt HM, Ebeling W, Rechenberg I, Schwefel H-P (eds) Parallel problem solving from nature, PPSN IV. Lecture Notes in Computer Sciences, vol 1141. Springer, Berlin, pp 245–254
44. Koziel S, Michalewicz Z (1999) Evolutionary Algorithms, Homomorphous Mappings, and Constrained Parameter Optimization. *Evol Comput* 7(1):19–44
45. Souza SA (2012) Optimal management of oil production using particle swarm optimization algorithm and adaptive surrogate models. Master thesis, Civil Engineering Department, UFPE, Recife, Brazil (in Portuguese)
46. Eldred MS, Giunta AA, Collis SS (2004) Second-order corrections for surrogate-based optimization with model hierarchies. Paper AIAA-2004-4457 in proceedings of the 10th AIAA/ISSMO multidisciplinary analysis and optimization conference, Albany, NY
47. CMG (2012) User's Guide. Computer Modeling Group Ltd., Calgary
48. Silva FT, Rodrigues JP, Paraizo PB, Romeu RK, Peres AM, Oliveira RM, Pinto IA, Maschio C (2005) Novel ways of parameterizing the history matching problem, SPE 94875, SPE Latin American and Caribbean Petroleum Engineering Conference, Rio de Janeiro