Particle swarm algorithm with adaptive constraint handling and integrated surrogate model for the management of petroleum fields

Innocente, MS, Afonso, SMB, Sienz, J & Davies, HM

Original citation & hyperlink:
Innocente, MS, Afonso, SMB, Sienz, J & Davies, HM 2015, 'Particle swarm algorithm with adaptive constraint handling and integrated surrogate model for the management of petroleum fields' Applied Soft Computing, vol 34, pp. 463-484. DOI: 10.1016/j.asoc.2015.05.032
https://dx.doi.org/10.1016/j.asoc.2015.05.032

DOI 10.1016/j.asoc.2015.05.032
ISSN 1568-4946
ESSN 1872-9681

Publisher: Elsevier

This manuscript has been accepted for publication in Applied Soft Computing. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all disclaimers that apply to the journal apply to this manuscript. A definitive version was subsequently published in *Applied Soft Computing, Vol.34, (2015), pp. 463–484*, doi: 10.1016/j.asoc.2015.05.032.

Particle Swarm Algorithm with Adaptive Constraint Handling and Integrated Surrogate Model for the Management of Petroleum Fields

Mauro Sebastián Innocente[a,*], Silvana Maria Bastos Afonso[b], Johann Sienz[a], Helen Margaret Davies[a]

[a] *Civil & Computational Engineering Centre, College of Engineering, Swansea University, Singleton Park, Swansea SA2 8PP, United Kingdom.*
[b] *Dept. Eng. Civil, Universidade Federal de Pernambuco, Recife-PE, Brazil.*

ABSTRACT

This paper deals with the development of effective techniques to automatically obtain the optimum management of petroleum fields aiming to increase the oil production during a given concession period of exploration. The optimization formulations of such a problem turn out to be highly multimodal, and may involve constraints. In this paper, we develop a robust Particle Swarm algorithm coupled with a novel adaptive constraint-handling technique to search for the global optimum of these formulations. However, this is a population-based method, which therefore requires a high number of evaluations of an objective function. Since the performance evaluation of a given management scheme requires a computationally expensive High-Fidelity simulation, it is not practicable to use it directly to guide the search. In order to overcome this drawback, a Kriging surrogate model is used, which is trained offline via evaluations of a High-Fidelity simulator on a number of sample points. The optimizer then seeks the optimum of the surrogate model.

Keywords: adaptive constraint handling; global search; Kriging; oil production; particle swarm; reservoir simulation; surrogate-based optimization; waterflooding management.

1. Introduction

The search for the optimum management scheme that maximizes oil production in petroleum fields is one of the major challenges in Petroleum Engineering. In this context, reservoir simulations and optimization methods are extensively used. Thus, Zhao et al. [1] use a *Simulated Annealing* based optimizer to determine the optimum steam injection pressure and steam-solvent flooding strategy in a thin heavy oil reservoir in the absence and presence of a bottom water zone.

Since the Net Present Value (NPV) is related to the production profit, it is commonly used in Reservoir Engineering management as the objective function [1], [2], [3], [4]. Waterflooding (WF) is the most widespread method used to improve oil recovery after primary depletion; i.e. after exhausting the reservoir's natural energy. The method consists of injecting water to raise the pressure and increase oil production. Horowitz et al. [2] propose four formulations of the WF management problem leading to optimization problems of different complexities, using the NPV as the function to be maximized subject to constraints at platform's total rates. They use a Sequential Approximate Optimization (SAO) procedure with a Sequential Quadratic Programming (SQP) local optimizer. This is a strategy proposed in [5], whose main feature is the sub-division of the original problem into a sequence of sub-problems to be solved in a sub-region of the original space named Trust Region (TR). Surrogate models to be called by the optimizer are built in the TR domain, which is updated as the search progresses [5], [6], [7]. This technique is also used by other researchers in the WF optimization context [3].

The concession period is usually subdivided into a number of control cycles with fixed switching times, with the well rates in each cycle set as design variables. Oliveira and Reynolds [4] present a hierarchical procedure to determine appropriate number and duration of control cycles. The well-by-well approach is based on criteria for refining/coarsening of control cycles based on gradients of the objective function and on differences between consecutive well controls at each well. If gradients are not available, only the latter criterion is applicable, in which case the merging potential may be affected if optimal controls tend to be rough.

While some of the formulations in [2] result in highly multimodal objective functions, the solutions found by the SQP optimizer are very sensitive to the initial guess. Hence we propose here to use a global search algorithm called Particle Swarm Optimization (PSO). The latter is composed of particles with different settings aimed at displaying complementary capabilities, and a so-called forward topology with time-increasing connectivity for the social network. In addition, since the plain PSO algorithm does not handle constraints on its own, an adaptive Constraint-Handling Technique (CHT) is developed and integrated into the optimizer. However, the fact that PSO is a population-based method also implies that it requires a high number of evaluations of the objective function. Given that the performance of a management scheme cannot be evaluated explicitly but by means of a computationally expensive High-Fidelity (HF) simulation, it is not feasible to use it directly to guide the search. Surrogate-based optimization has proved useful to the optimization of computationally expensive simulation-based models in the aerospace, automotive and oil industries [8]. Therefore, it is

* Corresponding author. Tel.: +44 (0)1792 606905
E-mail address: mauroinnocente@yahoo.com.ar

proposed in this paper that a Kriging surrogate Model (KM) [9], [10] be used, which is trained offline via evaluations of a HF commercial simulator (IMEX [11]) on a number of sample points. The selection for this technique is based on results from previous work in which several procedures for the surrogate construction of the NPV function were tested [2], [12].

The WF management problem is of high importance in Petroleum Engineering, whose objective is to increase productivity in petroleum fields using the rates of injector and producer wells as control parameters, thereby maximizing their economic return. In this paper, we show that a Particle Swarm (PS) algorithm with adaptive constraint handling and a static Kriging model can be combined to obtain near optimal results without the hassle of extensive numerical trial-and-error testing and tuning on a case-by-case basis. It is important to note that no tuning is carried out in this paper.

The layout of the paper is as follows: section 2 presents the WF problem general formulation and four alternatives according to the operational conditions; section 3 offers a discussion on surrogate models, in particular on Kriging approximations; section 4 presents an overview of the PSO method, emphasizing the features that are used in our code; section 5 presents the proposed PSO algorithm, including the formulation and settings of the particles' trajectory recurrence relation and neighbourhood topology, and the development of a novel adaptive CHT and termination conditions; section 6 presents the Proposed Integrated Tool (PIT), consisting of the tandem Kriging-PSO for the global surrogate-based optimization of the WF problem; finally, results from computational experiments are offered in section 7, a discussion of results is carried out in section 8, whilst conclusions and future work are presented in section 9.

2. Waterflooding Problem Formulation

The general formulation for the WF problem can be written as shown in Eq. (1):

$$\text{Maximize } NPV = f(\mathbf{q}) = \sum_{t=1}^{n_t}\left[\frac{1}{(1+d)^{\tau_t}}\cdot F(\mathbf{q}_t)\right]$$

$$\text{Subject to } \begin{cases} \sum_{p\in P} q_{p,t} \le Q_{l,\max} \\ \sum_{p\in I} q_{p,t} \le Q_{inj,\max} \\ q^l_{p,t} \le q_{p,t} \le q^u_{p,t} \;\; ; \;\; p = 1\ldots n_w \\ \sum_{p\in P} q_{p,t} \le \sum_{p\in I} q_{p,t} \le \delta \cdot \sum_{p\in P} q_{p,t} \\ t = 1\ldots n_t \end{cases} \quad (1)$$

where $\mathbf{q} = \left[\mathbf{q}_1^T\ \mathbf{q}_2^T \ldots \mathbf{q}_{n_t}^T\right]^T$ is the vector of well rates for all control cycles; $\mathbf{q}_t = \left[q_{1,t} \ldots q_{n_w,t}\right]^T$ 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 equation, d is the discount rate; τ_t is the time at the end of control cycle t; and $F(\mathbf{q}_t)$ is the cash flow at control cycle t, which represents the oil revenue minus the cost of water injection and water production. This is given by Eq. (2):

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

where $\Delta\tau_t$ is the time length of control cycle t; P and I distinguish producer from injector wells; $q^o_{p,t}$ and $q^w_{p,t}$ are the average oil and water rates at production well p at control cycle t; r_o is the oil price; and c_w and c_{wi} are the costs of producing and injecting water. In Eq. (1), $Q_{l,\max}$ 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 refer to the lower and upper bounds of design variables, respectively. Superscripts o and w denote oil and water phases, respectively. The last constraint in Eq. (1) requires that, for all cycles, the total injection rate belong to an interval that goes from the total production rate to δ times this value, where $\delta \ge 1$ is the over injection parameter. The commonly used approach to these problems is to subdivide the concession period into a number of control cycles, n_t, with fixed switching times. The design variables are the well rates in each control cycle. Four alternative formulations derived from Eq. (1) are proposed in [2], where they combine different platform operational conditions with and without the inclusion of the switching times of the control cycles as design variables. The operational conditions considered are:

1. Full Capacity Operation (FCO), in which the sum of both production and injection rates are at maximum platform's total rates. Under this assumption, the last equation presented in Eq. (1) is automatically satisfied. These equality constraints actually simplify the problem, as they result in variables expressed in terms of others, thus reducing the dimensionality of the search-space and removing those constraints from the formulation of the optimization problem.
2. Non-Full Capacity Operation (NFCO), in which the total injection and production rates may vary in order to increase the NPV, while the voidage replacement type constraints (last equation in Eq. (1)) are kept.

In this paper, situations where the control cycles are determined by the user are referred to as Fixed Time (FT) whereas those where the control cycles comprise design variables are referred to as Variable Time (VT). The cases resulting from the combination of operational conditions and types of switching times are depicted in Table 1. For each case, the number of design variables (n) and the type of constraints involved are shown. In the table, n_P is the number of producer wells and n_I is the number of injector wells. The mathematical formulation of each of these cases in Table 1 can be found in [2].

3. Surrogate Models

Surrogate models are built to provide smooth functions accurate enough to capture the general trends

Table 1
Characteristics of the four formulations of the WF problem. Equality constraints are not transferred to the formulation of the optimization problems but reduce dimensionality instead.

Problem	Dimensionality (n)	Constraint Type
FCO-FT	$(n_P + n_I - 2) \cdot n_t$	Side, Equality
NFCO-FT	$(n_P + n_I) \cdot n_t$	Side, Inequality
FCO-VT	$(n_P + n_I - 2) \cdot n_t + n_t - 1$	Side, Equality, Inequality
NFCO-VT	$(n_P + n_I) \cdot n_t + n_t - 1$	Side, Inequality

of the HF model at a considerably lower computational cost. These properties make them especially adequate for optimization purposes.

Surrogate models can be classified by differentiating *functional* from *physical* (hierarchical fidelity) categories [9]. The former comprises approaches such as data-fitting, polynomial series, and reduced-order methods, whereas the latter involves physical-based models. In this paper, since the performance of a given management scheme is evaluated by means of a commercial simulator acting as a black-box, a data-fitting type is adequate; in particular, a Kriging data-fitting model [9], [13], [14] is used. The central idea of this model is to assume that errors are not independent but rather exhibit spatial correlation related to the distance between corresponding points modelled 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 numerous peaks and valleys. Moreover, Kriging Models (KMs) provide exact interpolation at the sample points.

The first step in the construction of a KM is to generate the sampling points in the design space, which can be performed using Design of Experiments (DoE) Techniques [10], [15]. Latin Centroidal Voronoi Tessellation (LCVT) [16] is the DoE technique used in this paper. This selection is based on studies presented in [12], where an extensive surrogate model assessment considering different combinations of data-fitting and DoE techniques were conducted for the NPV response considered here. Once m sample points are generated, predictor expressions are developed to evaluate the function at untried design points.

In the KM literature, the true unknown function can be written as in Eq. (3):

$$f(\mathbf{x}) = \sum_{j=1}^{k} \beta_j \cdot N_j(\mathbf{x}) + Z(\mathbf{x}) \tag{3}$$

In the above equation, the first part is a linear regression of the data with k regressors, in which β_j are the unknowns and $Z(\mathbf{x})$ is the error. The latter is a (generally normally distributed) random function with zero mean, σ^2 variance, and non-zero covariance. The first term provides a global model or trend over the design space, while $Z(\mathbf{x})$ is responsible for creating a *localized* deviation from the *global* model. Polynomials are generally used to construct $N_j(\mathbf{x})$. A traditional approach called ordinary Kriging employs a zero-order (constant) function, so that the true unknown function becomes as in Eq. (4):

$$f(\mathbf{x}) = \beta + Z(\mathbf{x}) \tag{4}$$

where β is an unknown constant to be estimated based on m observed response values (samples). The covariance matrix of $Z(\mathbf{x})$ is given by Eq. (5):

$$Cov\left[Z(\mathbf{x}^i), Z(\mathbf{x}^j)\right] = \sigma^2 \cdot \mathbf{R} \cdot \left[R(\mathbf{x}^i, \mathbf{x}^j)\right] \tag{5}$$

where $\mathbf{x}^i$ and $\mathbf{x}^j$ are the sample points and $\mathbf{R}$ is the $m \times m$ correlation matrix of the stochastic process. Gaussian correlation function is used here. Therefore,

$$\mathbf{R}_{ij} = R(\mathbf{x}^i, \mathbf{x}^j) = \exp\left[-\sum_{k=1}^{n} \theta_k \cdot \left| \mathbf{x}_k^i - \mathbf{x}_k^j \right|^2 \right] \tag{6}$$

where θ_k is the k^{th} unknown correlation parameter used to fit the model.

Predicted estimates of the response $f(\mathbf{x})$ at any point of the design domain are defined as $\hat{f}(\mathbf{x}) = E(f(\mathbf{x})|\mathbf{f}_s)$, which stands for the expected value of $f(\mathbf{x})$ given the information in the true values of the function at m sampling points $\mathbf{f}_s = [f_1, \ldots, f_m]$.

A measure of the amount of error between the KM and the true model can be found introducing the concept of Mean Squared Error (MSE):

$$MSE = E\left(\hat{f}(\mathbf{x}) - f(\mathbf{x})\right)^2 \tag{7}$$

By minimizing the MSE in Eq. (7), the best unbiased predictor in [9] is obtained, as shown in Eq. (8):

$$\hat{f}(\mathbf{x}) = \hat{\beta} + \mathbf{r}(\mathbf{x})^T \cdot \mathbf{R}^{-1} \cdot \left(\mathbf{f}_s - \beta \cdot \mathbf{f}\right) \tag{8}$$

where $\hat{\beta}$ is unknown; $\mathbf{f} = [1, \ldots, 1]$ is a vector of ones; and $\mathbf{r}(\mathbf{x})$ is a correlation vector between untried $\mathbf{x}$ and the m sample data points, as shown in Eq. (9).

$$\mathbf{r}(\mathbf{x}) = \left[R(\mathbf{x}, \mathbf{x}^1), \ldots, R(\mathbf{x}, \mathbf{x}^m)\right] \tag{9}$$

As described in [9], the unknown parameters θ_k present at Gaussian correlation function R are found using maximum likelihood estimation, which is reduced to an n-dimensional minimization problem with simple bounds. In this approach, the values for $\hat{\beta}$ are obtained using generalized least squares, resulting in Eq. (10).

$$\hat{\beta} = \left(\mathbf{f}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{f}\right)^{-1} \cdot \mathbf{f}^T \cdot \mathbf{R}^{-1} \cdot \mathbf{f}_s \tag{10}$$

In KMs, some assessment strategies are required to check *a priori* if a generated model is adequate. They also provide guidelines for selecting the best surrogate when different options are constructed. The selection of the surrogate model in this paper is as in [12], which is based on results obtained considering two strategies: the *Root Mean Square Error* (RMSE); and the *Predicted Error Sum of Squares* (PRESS) [9].

4. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a global, population-based and gradient-free search method. In its standard form, it is suitable for single-objective, unconstrained problems with real-valued variables. Nevertheless, adaptations can be made to handle multiple objectives, constraints, and discrete problems [17], [18], [19], [20], [21], [22], [23], [24], [25].

The method was proposed by Kennedy and Eberhart [26], inspired by cooperative behaviour observed in social animals. While finding the global optimum is not guaranteed, it is a global optimizer in the sense that it is able to escape poor local attractors, which is possible due to a parallel search carried out by a swarm of cooperative particles sharing individually acquired information. The method does not necessarily return optimal solutions, as no optimality condition is considered. Nonetheless, it is a robust, general-purpose search method able to cope with problems for which it was not specifically designed or tuned. The method can also be viewed as a generator of good initial guesses for efficient local optimizers [27], [28]. For further reading, see [29], [30], [31], [32], [33].

4.1. Search Algorithm

The ability of the PSO method to optimize stems from decentralized local interactions among a swarm of particles. Its overall behaviour can be viewed as the overlap between each particle's individual behaviour and the social behaviour controlling the way individually acquired information is shared among particles [34]. The individual behaviour is governed by Eq. (11):

$$\begin{aligned} x_{ij}^{(t)} &= x_{ij}^{(t-1)} + \omega \cdot \left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)} \right) + \\ &+ i\phi \cdot \left(pb_{ij}^{(t-1)} - x_{ij}^{(t-1)} \right) + s\phi \cdot \left(pb_{kj}^{(t-1)} - x_{ij}^{(t-1)} \right) \end{aligned} \tag{11}$$

where $x_{ij}^{(t)}$ is the j^{th} coordinate of the position of particle i at time-step t; $i\phi$ and $s\phi$ are the individual and social acceleration coefficients, respectively; ω is the inertia weight; $pb_{ij}^{(t)}$ is the j^{th} coordinate of the best position found by particle i by time-step t; and $pb_{kj}^{(t)}$ is the j^{th} coordinate of the best position found by particle k, which is a neighbour of particle i, by time-step t. In classical formulations, neighbour k is the best-performing particle in the neighbourhood of particle i, and the individual and social acceleration coefficients are as in Eq. (12):

$$\begin{aligned} & i\phi = iw \cdot U_{(0,1)} = U_{(0,iw)} \\ & s\phi = sw \cdot U_{(0,1)} = U_{(0,sw)} \\ & 0 \le (\phi = i\phi + s\phi) \le (aw = iw + sw) \end{aligned} \tag{12}$$

where iw and sw are the individuality and sociality weights; aw is the acceleration weight; and $U_{(a,b)}$ is a random number from a uniform distribution in $[a,b]$.

The settings of the coefficients in the trajectory recurrence relation greatly influence the behaviour of the system. Pioneering work on the subject was carried out by Kennedy [35]; Ozcan et al. [36]; Clerc et al. [37]; Trelea [38]; and van den Bergh et al. [39]. More recently, Innocente et al. [40] studied the effect of the settings of ω and ϕ on the speed and form of convergence of a deterministic particle pulled by a stationary attractor (**p**). They propose a more general formulation aiming to control the degree of randomness in ϕ, where $i\phi$ and $s\phi$ are as in Eq. (13) (see also [33]):

$$\begin{aligned} & i\phi = \iota \cdot \left[\phi_{\min} + (\phi_{\max} - \phi_{\min}) \cdot U_{(0,1)} \right] \\ & s\phi = (1-\iota) \cdot \left[\phi_{\min} + (\phi_{\max} - \phi_{\min}) \cdot U_{(0,1)} \right] \\ & \iota \in [0,1) \end{aligned} \tag{13}$$

Within this framework, the classical formulation is given by $\phi_{\min} = 0$ and $\iota = 0.5$, whereas the popular Constricted Original PSO (COPSO) formulation of Clerc et al. [37] is as in Eq. (14):

$$\begin{aligned} & \omega = \begin{cases} \dfrac{2 \cdot \kappa}{aw - 2 + \sqrt{aw^2 - 4 \cdot aw}} & \text{if } aw \ge 4 \\ \kappa & \text{otherwise} \end{cases} \\ & \kappa \in (0,1); \quad \phi_{\min} = 0; \quad \phi_{\max} = \omega \cdot aw; \quad \iota = 0.5 \end{aligned} \tag{14}$$

where the user typically sets aw close to but greater than 4, and κ close to but smaller than 1.

Other authors focus on the development of automatic tuning of these coefficients. Thus, Nickabadi et al. [41] identify two situations for a particle during the search:

1. The particle and the global best position are far from the optimum, and the particle's displacement sizes are low compared to its distance to the optimum.
2. The global best position is close to the optimum and the particle's position is far from them.

They argue that the first case requires high values of ω because the particle is *exploring*, whereas low values of ω are required in the second case because the particle is exploiting previously found promising regions. They propose using ω as an adaptive parameter to control the exploratory and exploitative behaviour of the swarm.

Leu et al. [42] propose two parameter automation strategies, one for ω and the other for the acceleration coefficients. Thus, ω and sw are updated adaptively using grey relational analysis with the best particle as reference, and $iw = 4 - sw$. The adaptive rules consist of linear functions of the grey relational grade between predefined bounds, where the higher the grade the lower the ω and the higher the sw. This is because higher grades imply higher similarity and hence exploitation; i.e. they argue that lower ω and higher sw for constant $sw + iw$ are desirable during exploitation. Aiming to fix some weaknesses of the grey PSO, Leu et al. [43] propose using grey evolutionary analysis. In this case, the adaptation is based on an evolutionary factor computed using a modified version of the grey relational grade, where the distribution of particles at all previous time-steps is considered. In this approach, both ω and sw increase with the evolutionary factor, with the increase being nonlinear and $iw = 4 - sw$. Thus, they consider that ω should be increased during exploitation.

Loosely speaking, a particle's individual behaviour is controlled by the coefficients in its trajectory recurrence relation; i.e. ω, sw and iw in Eqs. (11) and (12), or ω, $\phi_{\min}$, $\phi_{\max}$ and ι in Eqs. (11) and (13). In fact, it only depends on ω and ϕ, as shown later in Eq. (16).

In turn, particles interact locally by exchanging individually acquired information. This exchange is

indirect because a particle accesses information from the memories of neighbouring particles without keeping this information in its own memory but having its trajectory influenced by it. This local exchange is controlled by a local sociometry that defines which memories the particle can access. Thus, a particle's social behaviour materializes through the update of its social attractor ($\mathbf{pb}_k$ in Eq. (11)), and also through the stochastically weighted average between its social and its individual attractors ($\mathbf{pb}_k$, $\mathbf{pb}_i$) to generate its overall attractor ($\mathbf{p}_i$):

$$p_{ij}^{(t)} = \frac{i\phi \cdot pb_{ij}^{(t)} + s\phi \cdot pb_{kj}^{(t)}}{i\phi + s\phi} = \frac{i\phi \cdot pb_{ij}^{(t)} + s\phi \cdot pb_{kj}^{(t)}}{\phi}. \quad (15)$$

Thus, without loss of generality:

$$x_{ij}^{(t)} = x_{ij}^{(t-1)} + \omega \cdot \left(x_{ij}^{(t-1)} - x_{ij}^{(t-2)}\right) + \phi \cdot \left(p_{ij}^{(t-1)} - x_{ij}^{(t-1)}\right). \quad (16)$$

The social behaviour of the swarm as a whole emerges from the individual behaviour of every particle combined with their local interactions and with features defined at the global level such as the synchrony of the updates. The overall sociometry results from assembling all local sociometries. For further reading on the subject, refer to [44], [45], [46], [30], [47], [48], [49] and [33].

Hence the convergence speed and the ability of PSO to escape poor suboptimal solutions are controlled by the convergence speed of each particle and by the speed at which information is spread throughout the swarm.

In general, we refer to the function to be optimized, $f(\mathbf{x})$, as the *conflict function* due to the original metaphor that individuals seek agreement by minimizing their *conflicts* in the space of beliefs [50], [29]. The more general denomination of *objective function* may also be used indistinctively. Other popular names are *fitness function*, imported from Evolutionary Optimization, and *cost function*, imported from the minimization of cost curves in Operations Research. Similarly, $\mathbf{x}$ is the vector of *objective variables*, also referred to as *problem variables*, *decision variables*, or *design variables*.

4.2. Constraint-Handling

Constraints bound the regions of the search-space where solutions are admissible. Some optimization methods such as the Simplex Method or SQP are inherently constrained optimization methods in the sense that the constraint-handling is embedded in the search strategy. Conversely, there is nothing in the standard PSO algorithm that tells particles how to handle constraints. Hence some Constraint-Handling Technique (CHT) must be incorporated for constrained problems.

The constrained optimization problem is formulated as in Eq. (17) for convenience in PSO:

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{Subject to} \\ &\begin{cases} g_j(\mathbf{x}) \le Tol_{ineq} & ; \quad j = 1, \ldots, ni \\ \text{abs}\left(g_j(\mathbf{x})\right) \le Tol_{eq} & ; \quad j = ni+1, \ldots, ni+ne \\ \max(0, x_i - u_i) + \max(0, -x_i + l_i) \le 0 \\ i = 1, \ldots, n \end{cases} \end{aligned} \quad (17)$$

where $f(\mathbf{x})$ is the *objective function*; $g_j(\mathbf{x})$ is the j^{th} constraint function; n is the dimensionality of the problem; ni is the number of inequality constraints; ne is the number of equality constraints; l_i and u_i are the lower and upper bounds for the i^{th} dimension (variable), respectively; $Tol_{ineq} \ge 0$ is the tolerance for inequality constraints violations; and $Tol_{eq} > 0$ is the tolerance for equality constraints violations.

A straightforward CHT is the penalization method, where infeasible solutions are penalized by increasing their objective function values and treating the problem as unconstrained. The general rule is that the amount of penalization be linked to the amount of constraint violation. Other CHTs consist of introducing adaptation in the penalization [51]; using augmented Lagrange Multipliers [52]; considering the objective function values and the constraint violations separately [53]; or formulating constraints as additional objectives [21], [20]. For a survey of CHTs, refer to [54].

4.3. Applications

Not only can PSO be applied directly to optimization problems, but also to a wide range of industrial problems that can be posed as such. Examples of the latter are structural design [52]; scheduling [55]; engineering optimal design [51]; shortest path problems [18]; neural networks training [56], [57]; data classification [58]; real-time moving object tracking [59]; reactive power dispatch [60]; software testing [61]; etc. For reviews of PSO applications, refer to [62], [63] and [64].

5. Proposed Optimizer

Given the strong multimodality resulting from the formulations of the WF management problem, a global optimization algorithm for real-valued variables would be appropriate to seek the management scheme that maximizes oil production. There is a plethora of global search methods available in the literature, among which some of the most popular and widely tested ones are Genetic Algorithms (GAs), Evolution Strategies (ESs), Differential Evolution (DE), and PSO. These and other nature-inspired global search methods have a number of settings that affect their behaviour. Moreover, there are uncountable variants of each one of them, as well as hybridizations among them and also with classical local search methods. Therefore, it is not possible to identify a best problem-solver matching at the paradigm level. At most, a few specific variants with specific settings could be compared on a specific set of problems with specific characteristics and dimensionality. Clearly, this paper is not aimed at such comparisons.

PSO and DE have gained increasing interest for a couple of decades, as numerous successful applications have been reported [62], [64], [65], [66]. Swagatam et al. [65] provided an overview of these two algorithms, which they claim are currently gaining popularity for their greater accuracy, faster convergence speed and simplicity. In particular, many successful applications of PSO have been reported in which this algorithm has shown advantages over other nature-inspired algorithms, mainly due to its robustness, efficiency and simplicity

[67], [63], [68], [69]. Thus, while the use of other global search methods is certainly feasible, we propose the use of a PSO algorithm to search for a near-optimal solution to the multimodal WF management problem. It is not argued that it is the unquestionably best alternative, but simply that the method performs well on these kinds of problems, where a good candidate solution generally implies that there is a better one in its proximity.

5.1. Search Algorithm

Swarm sizes in PSO commonly vary from 10 to 20 particles for simple problems, and from 20 to 100 for complex ones. It can be reported that under testing, no swarm size between 20 to 100 particles produced results that were clearly superior or inferior to any other value for a majority of the tested problems [70]. A swarm of 50 particles is used in this paper, the same as in [70].

The speed and form of convergence of the proposed PSO search algorithm is controlled by the settings of the coefficients in Eqs. (11) and (13), combined with the topology of the neighbourhood. Note that, as opposed to classical formulations, $\phi_{\min} \neq 0$ is allowed in our formulations. The settings of these coefficients affect the way a particle searches around its attractor, having an influence on its convergence speed as well as on the amplitudes and frequencies in its trajectory oscillations.

Typically, all particles are identical, having the same coefficients' settings. We propose to divide the swarm in sub-swarms of particles displaying different individual behaviours; i.e. each sub-swarm has different settings of its coefficients. The aim is to complement their abilities to cope with a range of environments posing different difficulties, thus becoming more general-purpose. This strategy is supported by experiments in [33].

Given that cooperation between less than 10 to 15 particles may not be sufficient for complex problems, we divide our swarm of 50 particles into three sub-swarms of 17, 16 and 17 particles, respectively. While Liang and Suganthan [71] used more and smaller sub-swarms, they implemented a re-grouping mechanism to re-introduce diversity and allow cooperation among sub-swarms. Blackwell and Branke [72] also proposed the use of sub-swarms, but with two different interaction mechanisms, namely a local interaction mechanism between colliding sub-swarms called *exclusion*, and an information sharing interaction mechanism called *anti-convergence*.

For one sub-swarm, we use the popular Constricted Original PSO (COPSO), which introduces the *Type 1"* constriction factor (χ) proposed in [37]. The coefficients' settings are as recommended in [37]: $\kappa = 0.99994$ (they actually recommend $\kappa = 1$) and $aw = 4.1$, which results in $\chi = 0.7298$ (see Eq. (14) and settings in [70] and [73]). Translating these settings into our formulation, $\omega = 0.7298$, $\phi_{\min} = 0$, $\phi_{\max} = 2.9922$, and $\iota = 0.5$. For the other two sub-swarms, we propose using the *Behaviour Type 1* and *Behaviour Type 2* formulations discussed in the next two sections. Recall that a particle i is in fact pulled by a single attractor $\mathbf{p}_i^{(t)}$, as shown in Eqs. (15) and (16).

5.1.1. Behaviour Type 1

Here we study the average behaviour (i.e. $\phi = \phi_{\text{mean}}$) of a particle pulled by a stationary attractor $\mathbf{p}$. Hence the position variables can be expressed in vectorial form (boldface), and sub-index j in Eq. (16) can be dropped. Given that we are studying an individual particle, sub-index i in Eq. (16) can also be dropped.

In line with our formulation in Eq. (13), the idea is to study the average individual behaviour of a given particle between updates of its attractor ($\mathbf{p}$), while stochasticity would be introduced as accumulative noise for each dimension independently (not analyzed here).

The aim is to find a relationship between ω and ϕ_{mean} to *cancel the momentum* once the particle has overflown the attractor. That is, if the particle overflies the attractor from time-step $(t-1)$ to time-step t, then $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)}$ so that the particle does not keep flying away from $\mathbf{p}$ in the next time-step. While it is not desirable in a deterministic algorithm to re-evaluate the same position, randomness would ensure that $\mathbf{x}^{(t+1)} \neq \mathbf{x}^{(t)}$. Thus, when the attractor is updated at $(t-1)$, the conditions imposed for the mean behaviour are offered in Eq. (18), where it is assumed that $\mathbf{x}^{(t-1)} = \mathbf{x}^{(t-2)}$ while $\phi_{\text{mean}} > 1$ ensures that the particle overflies the attractor from $(t-1)$ to t.

$$\begin{cases} \mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} + \phi_{\text{mean}} \cdot \left(\mathbf{p} - \mathbf{x}^{(t-1)}\right) \\ \mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} \\ \phi_{\text{mean}} > 1 \end{cases} \tag{18}$$

Since this analysis is between updates of $\mathbf{p}$, it is reasonable to assume that $\mathbf{x}^{(t-1)} = \mathbf{x}^{(t-2)}$ when an update takes place at time-step $(t-1)$, and therefore the particle does not have an initial momentum. This means that the particle converged to its previous attractor by the time a new attractor is found at $(t-1)$, starting its oscillatory trajectory towards it with $\mathbf{x}^{(t-1)} = \mathbf{x}^{(t-2)}$. Even if the particle did not converge to its previous attractor, it is reasonable to assume that its momentum would be small for convergent settings of $(\omega, \phi_{\text{mean}})$ (hence $\mathbf{x}^{(t-1)} \approx \mathbf{x}^{(t-2)}$). Thus, after some arithmetic manipulations of Eq. (18):

$$\begin{aligned} &\mathbf{x}^{(t+1)} - \mathbf{x}^{(t)} = \omega \cdot \left(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}\right) + \phi_{\text{mean}} \cdot \left(\mathbf{p} - \mathbf{x}^{(t)}\right) = 0 \\ &\omega \cdot \left(\mathbf{p} - \mathbf{x}^{(t-1)}\right) = -\mathbf{p} + \mathbf{x}^{(t)} \end{aligned} \tag{19}$$

Operating with $\mathbf{x}^{(t)}$ in Eqs. (18) and (19), we obtain:

$$\boxed{\omega = \phi_{\text{mean}} - 1} \tag{20}$$

Thus, starting from an update of the attractor at $t = 1$ with $\mathbf{x}^{(1)} = \mathbf{x}^{(0)}$ (no momentum) and setting $\omega \in (0,1)$ and ϕ_{mean} such that Eq. (20) holds, the average behaviour of the particle consists of overflying the attractor at $t = 2$, losing its momentum thus maintaining its position at $t = 3$, overflying the attractor again at $t = 4$, losing its momentum again and maintaining its position at $t = 5$, etc. Since the condition imposed for $\mathbf{x}^{(t+1)}$ in Eq. (18) is the same as the no momentum assumption at the beginning of the analysis ($\mathbf{x}^{(t-1)} = \mathbf{x}^{(t-2)}$), this pattern repeats until the particle converges.

An interesting case can be observed by extending the behaviour to $\phi_{\text{mean}} = 1$ and hence $\omega = 0$, thus removing the assumption of $\phi_{\text{mean}} > 1$: the particle converges in a single time-step. In turn, by extending the behaviour to $\phi_{\text{mean}} < 1$ and $\omega \in (-1,0)$, the particle still maintains its previous position every other time-step, but converges monotonically (i.e. the attractor is not overflown).

The convergence conditions for the deterministic trajectory of a particle are [34], [40], [33]:

1) $\omega < 1$
2) $\phi > 0$
3) $\phi < 2 \cdot (\omega + 1)$

All settings of (ω,ϕ) which satisfy these conditions result in convergent deterministic trajectories. The (ω,ϕ) pairs which satisfy $\omega \in [0,1]$ and Eq. (20) are shown in a red dotted line in Fig. 1 (top), where the region inside the black dotted parabola is the so-called complex region where the roots of the characteristic polynomial of the recurrence relation are complex. Within this region, the convergence speed of a particle's deterministic trajectory is proportional to $\omega^{0.5}$ [40], [33]. As can be observed, the segment line of the (ω,ϕ) pairs satisfying $\omega \in (0,1)$ and Eq. (20) is within convergent complex region.

We define the PSO-RRR1 (with RRR standing for Reduced Randomness Range) as a formulation which displays *Behaviour Type 1* and introduces stochasticity by defining ϕ_{min} as the average between ϕ_{mean} and the left convergence boundary, and ϕ_{max} as the average between ϕ_{mean} and the right convergence boundary, as in Eq. (21).

$$\begin{aligned} \phi_{mean} &= \omega + 1 \\ \phi_{min} &= \frac{1}{2} \cdot (\omega + 1) \\ \phi_{max} &= \frac{3}{2} \cdot (\omega + 1) \end{aligned} \tag{21}$$

As can be observed in Fig. 1 (top), the whole range of ϕ is within complex region for $\omega > 0.072$.

The deterministic trajectories corresponding to a set of settings with $\omega \in [0,1]$ and ϕ as in Eq. (20) in one dimensional space are shown in Fig. 2 (centre column), where $x^{(0)} = 100$ and $p = 0$.

5.1.2. *Behaviour Type 2*

Similar to the previous analysis, the aim now is to reach the attractor $(\mathbf{p})$ in two time-steps from the moment the attractor is updated at time-step $(t-1)$, also starting from stagnation (i.e. $\mathbf{x}^{(t-1)} = \mathbf{x}^{(t-2)}$).

$$\begin{cases} \mathbf{p} - \mathbf{x}^{(t-1)} = \mathbf{d} \\ \mathbf{x}^{(t+1)} - \mathbf{x}^{(t-1)} = \mathbf{d} \end{cases} \tag{22}$$

$$\mathbf{x}^{(t)} = \mathbf{x}^{(t-1)} + \phi \cdot \left(\mathbf{p} - \mathbf{x}^{(t-1)}\right) = \mathbf{x}^{(t-1)} + \phi \cdot \mathbf{d} \tag{23}$$

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \omega \cdot \left(\mathbf{x}^{(t)} - \mathbf{x}^{(t-1)}\right) + \phi \cdot \left(\mathbf{p} - \mathbf{x}^{(t)}\right) \tag{24}$$

From (23) and (24),

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t-1)} + \phi \cdot \mathbf{d} + \omega \cdot \phi \cdot \mathbf{d} + \phi \cdot \left(\mathbf{p} - \mathbf{x}^{(t-1)} - \phi \cdot \mathbf{d}\right) \tag{25}$$

$$\boxed{\begin{aligned} \omega &= \frac{1}{\phi} - 2 + \phi \\ \phi &= \frac{2 + \omega \pm \sqrt{(2+\omega)^2 - 4}}{2} \end{aligned}} \tag{26}$$

The (ω,ϕ) pairs which satisfy $\omega \in [0,1]$ and Eq. (26)

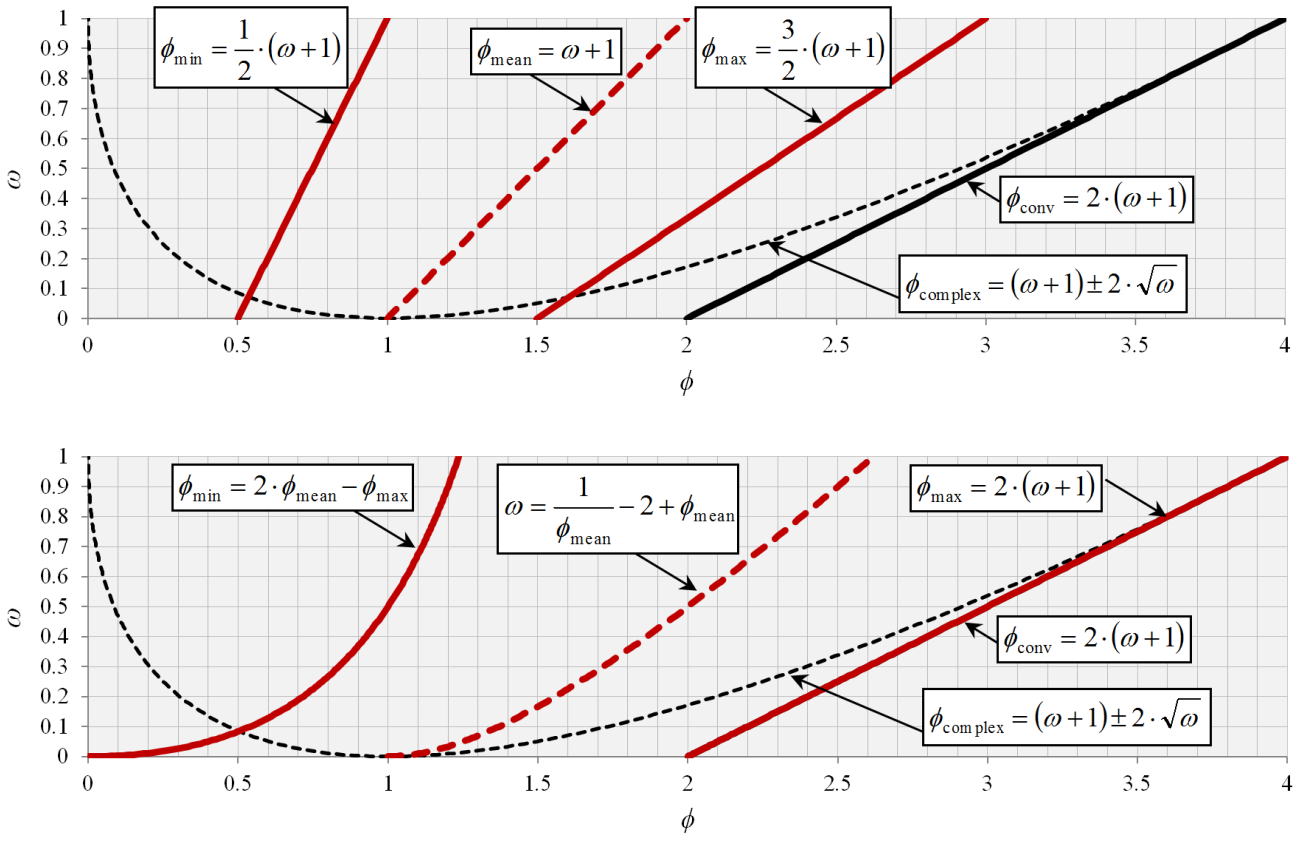

Fig. 1. Values of (ω,ϕ_{mean}) for *Behaviour Type 1* (top) and *Behaviour Type 2* (bottom) in the (ω,ϕ) plane (red, dotted lines). The figures also provide ϕ_{min} and ϕ_{max} for a given $\omega > 0$ and $\phi_{mean} > 1$ for PSO-RRR1 (top) and PSO-RRR2 (bottom).

with the positive square root ($\phi \geq 1$) are shown in a red dotted line in Fig. 1 (bottom). As can be observed, the segment line of the (ω,ϕ) pairs satisfying $\omega \in (0,1)$ and Eq. (26) is within convergent complex region.

We define the PSO-RRR2 as a formulation which displays *Behaviour Type 2* on the right branch, and introduces stochasticity by defining $\phi_{\max}$ on the right boundary of the convergence region and $\phi_{\min}$ accordingly. Thus, PSO-RRR2 is defined as in Eq. (27).

$$\begin{aligned} \phi_{\text{mean}} &= \frac{2+\omega+\sqrt{(2+\omega)^2-4}}{2} \\ \phi_{\max} &= 2\cdot(\omega+1) \\ \phi_{\min} &= 2\cdot\phi_{\text{mean}}-\phi_{\max} \end{aligned} \tag{27}$$

As shown in Fig. 1 (bottom), the whole range of ϕ is not kept within complex region for PSO-RRR2.

The deterministic trajectories corresponding to a set of settings with $\omega \in [0,1]$ and ϕ as in Eq. (26) in one dimensional space are shown in Fig. 2 (left and right columns), where $x^{(0)} = 100$ and $p = 0$. Trajectories on the left column are for $\phi \in (0,1]$ (left branch) whereas the ones on the right column are for $\phi \geq 1$ (right branch). Trajectories on the same row display the same convergence speed [33]. Recall that PSO-RRR2 requires $\phi_{\text{mean}} > 1$ (right column only).

As opposed to *Behaviour Type 1*, the condition imposed for $\mathbf{x}^{(t+1)}$ in Eq. (22) ($\mathbf{x}^{(t+1)} = \mathbf{p}$) is unrelated to the no momentum assumption at the beginning of the analysis ($\mathbf{x}^{(t-1)} = \mathbf{x}^{(t-2)}$). Hence the pattern of having no

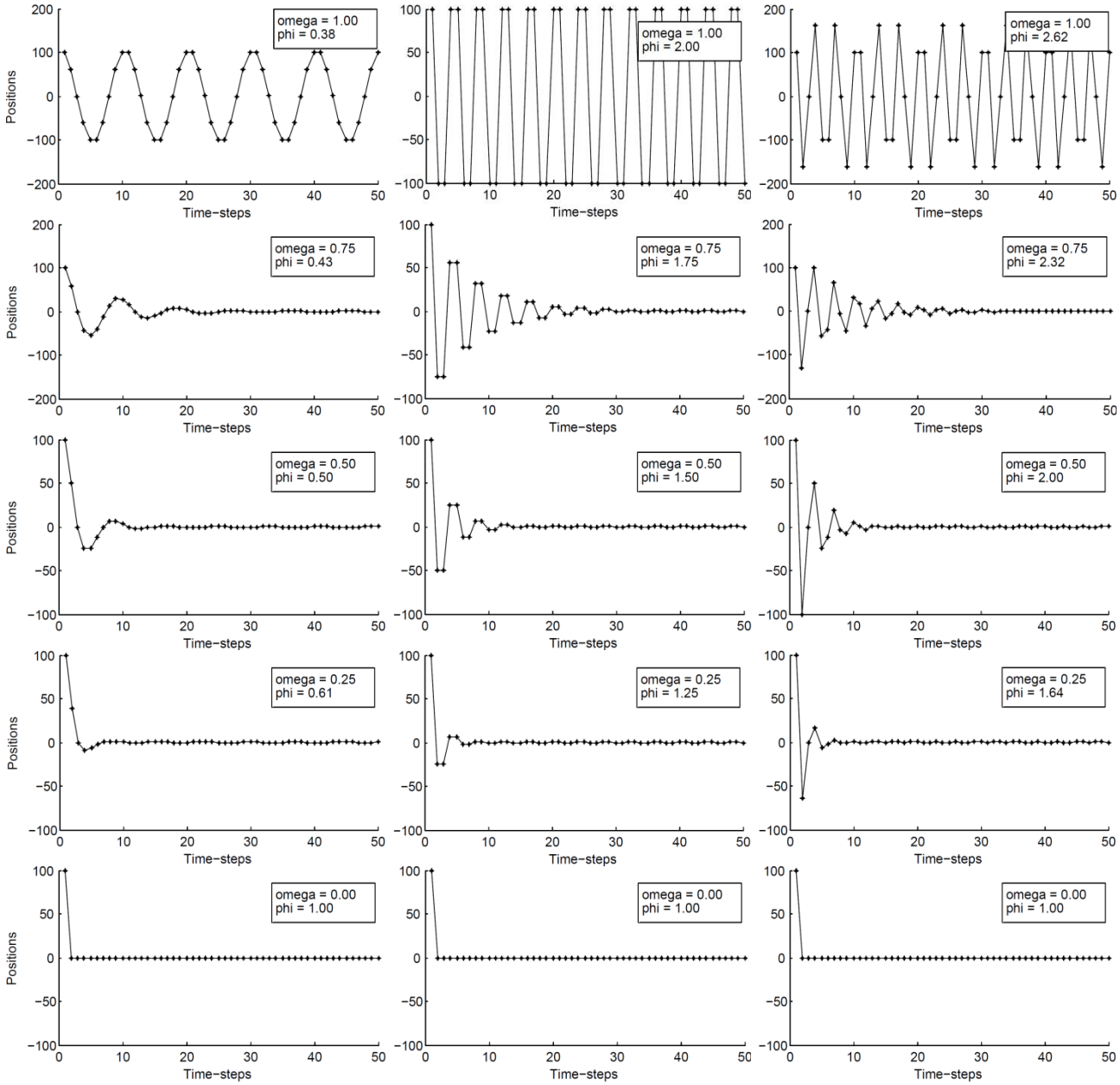

Fig. 2. Deterministic trajectories corresponding to a set of settings along the lines displaying *Behaviour Type 1* (centre column), *Behaviour Type 2* with $\phi \leq 1$ (left column), and *Behaviour Type 2* with $\phi \geq 1$ (right column) for a particle in one dimensional space, where $x^{(0)} = 100$ and $p = 0$. Note the change of scale in the vertical axis for clarity in trajectories of *Behaviour Type 2* with $\omega = 1.00$ and $\omega = 0.75$. Trajectories in the same row display the same convergence speed. All three columns converge to (ω,ϕ) = (0,1).

momentum at time-step t and reaching the attractor at time-step $(t + 2)$ does not repeat. Thus, starting from an update of the attractor at $t = 1$ with $\mathbf{x}^{(1)} = \mathbf{x}^{(0)}$, $\omega \in (0,1)$ and ϕ_{mean} such that Eq. (26) holds, the average behaviour of the particle consists of reaching the attractor at $t = 3$, with the momentum taking it away from $\mathbf{p}$ at $t = 4$. From then on, the particle converges in a pattern unpredicted by this analysis, with high frequencies ($\phi_{mean} > 1$).

Trajectories resulting from *Behaviour Type 1* and *Behaviour Type 2* can be compared by observing the trajectories on the same row in the centre and right columns in Fig. 2, which share the same convergence speed. Note that both behaviour types converge on $(\omega,\phi) = (0,1)$, which results in maximum convergence speed.

Thus, our optimizer uses three sub-swarms:

1) PSO-RRR2 formulation, with $\omega = 0.8167$.
2) PSO-RRR1 formulation, with $\omega = 0.80$.
3) COPSO formulation, with $(\omega,\phi_{max}) = (0.7298, 2.9922)$.

The first sub-swarm displays the highest frequency and the largest amplitudes in the oscillations, a medium range of variation of ϕ ($\phi_{max} - \phi_{min} = 2.4667$) and it is the only one whose ϕ can reach the right boundary of the convergence region. The second sub-swarm displays medium range frequency in the oscillations, and the lowest range of variation of ϕ ($\phi_{max} - \phi_{min} = 1.8$). In turn, the third sub-swarm displays the lowest frequency in the oscillations (even though it has the lowest ω) and has the largest range of variation of ϕ ($\phi_{max} - \phi_{min} = 2.9922$).

These sub-swarms are independent from one another, interacting only by means of an information sharing mechanism different from those in [71] and [72], which is discussed in the next section.

5.1.3. Neighbourhood Topology

A so-called *forward topology* is proposed, which shares important characteristics with the classic *ring topology*. Namely, it allows any number of neighbours (*nn*) from 0 (no cooperation) to (swarm-size – 1) (full cooperation), and the graph that represents it is connected for any $nn > 0$. The difference between the *ring* and the *forward topologies* is that interconnections are not bidirectional in the latter, so that a particle is not generally informed by the same particles it informs.

A graphical comparison between the *ring* and the *forward topologies* is offered in Fig. 3 for 6 particles and 2 neighbours. Notice that the number of edges in the graphs that need to be traversed to go from a given node to its farthest node are the same in both cases. For instance, the farthest node for node 1 in Fig. 3 is node 4 for the *ring topology* (traversing nodes 2 and 3) and node 6 for the *forward topology* (traversing nodes 3 and 5). The reverse is only true for the *ring topology*: the farthest node for node 4 is node 1 for the *ring topology*, whereas the one for node 6 in the *forward topology* is not node 1 –which is actually the closest– but node 5.

In our proposed optimizer, an independent *forward topology* is used within each of the 3 sub-swarms, where the number of neighbours is time-increasing linearly from 1 at the first time-step (i.e. the neighbourhood of a particle is composed of 1 neighbour plus the particle itself) until it becomes global when the search reaches the maximum number of time-steps permitted (t_{max}).

The interaction between these 3 *forward topologies* takes place by extending the local sociometry of the first particle in each sub-swarm through their access to the individual memory of the other two.

The use of the forward topology with linearly time-increasing connectivity and of this so-called *individual overlapping* for the interaction between sub-swarms is supported by experimental results in [33].

5.2. Constraint-Handling

Some of the formulations for the WF management problem involve constraints. Therefore, a novel, adaptive CHT is developed and integrated into the optimizer.

We make use here of a Preserving Feasibility with Priority Rules (PFPR) technique, which consists of rules to decide, whenever two candidate solutions are compared, which one is better. The rules are as follows:

1) If they are both feasible, the one with the lowest objective function value is better.
2) If they are both infeasible, the one with the lowest Constraints Violation (CV) is better.
3) If one is feasible and the other infeasible, the feasible one is always better.
4) If they are both infeasible with the same amount of Constraint Violation (CV), the one with the lowest objective function value is better.

Similar CHTs have been proposed in the literature [53], [74], [19], [75]. Takahama et al. [19] relaxed these priority rules by means of a control parameter in the comparisons (the so-called ε-level comparisons).

The PFPR technique has the advantage that an initial feasible swarm is not required and that the objective function is seldom evaluated for infeasible particles. However, when searching highly constrained spaces, most of the search is driven by constraint satisfaction, disregarding the conflict function information. Thus, by the time a particle finds a feasible location, it might be anywhere with respect to the optimum.

5.2.1. Self-Tuned Initial Tolerances Relaxation

Since the use of a tolerance for equality CVs in PSO is a must, it is common practice to relax this tolerance at the beginning and decrease it as the search progresses. Usually, the tolerance is relaxed to an arbitrary initial

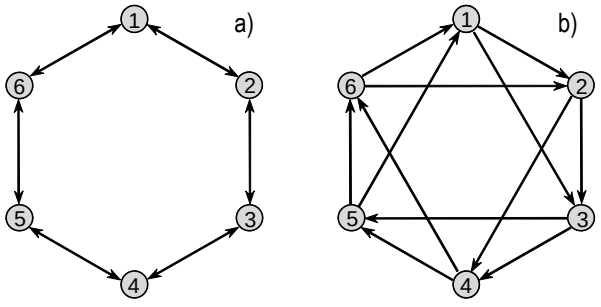

Fig. 3. Swarm composed of 6 particles, with a) *ring topology* with 2 neighbours; b) *forward topology* with 2 neighbours.

value, and then deterministically decreased. The aim is to temporarily expand the feasible region of the search-space to relax the priority rules in a similar fashion as the ε-level comparisons do. However, the impact of a given relaxation on the Feasibility Ratio (FR) of the search-space is problem-dependent, and can vary greatly. For instance, to obtain a FR $\in$ [0.20,0.25], a tolerance for equality CVs of around 0.26 is required for problem g11 in [33], [76], [53], whereas a tolerance of around 6.63 is required for problem g13 (both with equality constraints only). Since there are problems involving only inequality constraints which present small FRs, the same concept can be applied. That is, the tolerance for their violations can also be relaxed. Thus, the tolerance required for problem g10 in [33], [76], [53] to present a FR $\in$ [0.20,0.25] is around 10.83 whereas it is around 2790 for problem g06 (both with inequality constraints only).

Therefore, initial self-tuned tolerance relaxations are proposed aiming for a *target FR*. Here, we arbitrarily set *target FR* $\in$ [0.20,0.25]. Thus, the self-tuning procedure consists of starting with small, minimum values for the tolerances (0.01 for inequality and 0.1 for equality constraints), and evaluating the constraint functions on 1000 randomly selected positions. The FR is calculated, and the tolerances are adequately increased or decreased. More precisely, if the resulting FR < 0.20, the tolerances are increased by a factor for 10 whereas, if the resulting FR > 0.25, a bisection search is triggered to find the minimum tolerances which satisfy FR $\in$ [0.20,0.25]. For problems with FRs > 0.25, *target FR* $\in$ [(1.1 · FR),(1.1 · FR + 0.05)], with the obvious limit of 1.

5.2.2. Adaptive Decrease of Tolerances Relaxation

The aim is to make the tolerance update adaptive so that updates are performed when they would have a less disruptive effect on the dynamics of the swarm and on maintaining potentially good solutions. Thus, updates are performed when a given minimum percentage ($ptg_{\min}$) of the particles' best experiences (**pb**) are located within feasible space for current tolerances. The coefficient for the exponential update ($ktol^{(t)}$) is also adaptive, as shown in Eq. (28), while the exponential update of the tolerances is as posed in Eq. (29).

$$ktol^{(t)} = \frac{0.99 - ktol_{\min}}{100 - ptg_{\min}} \cdot (100 - ptg) + ktol_{\min} \tag{28}$$

$$Tol^{(t)} = ktol^{(t)} \cdot Tol^{(t-1)} \tag{29}$$

Thus, $ktol^{(t)} = 0.99$ for $ptg = ptg_{\min}$, $ktol^{(t)} = ktol_{\min}$ for $ptg = 100$, with linear variation in between. Therefore, the greater the percentage above a minimum established, the greater the size of the tolerance decrease.

Aiming to avoid too many time-steps without a tolerance update, a safety mechanism is implemented by enforcing an update if:

$$\frac{t}{ntu} \geq \Delta t \tag{30}$$

where t is the current time-step, ntu is the number of tolerances updates, and Δt is the maximum permitted average number of time-steps between tolerance updates. When an update is enforced by Eq. (30), the coefficient used in Eq. (29) is $ktol^{(t)} = 0.99$ (i.e. $ktol_{\max}$).

In order to give some time for the particles to find feasible solutions once the tolerances have reached their desired values, it is arbitrarily set that such values must be reached by the time 80% of the maximum search-length ($t_{\max}$) has elapsed. If the desired tolerances are not reached adaptively by the time 72% of $t_{\max}$ has elapsed, a tolerance update is enforced at every time-step so that the desired values are attained at $t = (0.80 \cdot t_{\max})$. Hence $ktol$ is calculated as in Eqs. (31) and (32), and is kept constant for the remaining $(0.08 \cdot t_{\max})$ time-steps.

$$ktol^{0.08 \cdot t_{\max}} \cdot Tol^{(0.72 \cdot t_{\max})} = Tol^{(0.8 \cdot t_{\max})} = Tol_{desired} \tag{31}$$

$$ktol = \left(\frac{Tol_{desired}}{Tol^{(0.72 \cdot t_{\max})}} \right)^{\frac{1}{0.08 \cdot t_{\max}}} \tag{32}$$

where $ktol$ is calculated independently for inequality and equality constraints.

For inequality CVs, $Tol_{desired}$ is typically set to 0. Since this cannot be reached by exponential updates, we set $Tol_{desired} = 10^{-5}$ for the calculation of $ktol$ in Eq. (32), re-setting the tolerance to 0 as soon as it reaches a value equal to or below 10^{-5} in Eq. (29).

5.2.3. Adaptive PFPR Technique

The proposed adaptive scheme is coupled with the PFPR technique, which is especially useful in problems with low FRs. By means of the adaptive scheme, the PFPR technique is fooled into using objective function information while searching infeasible space.

Thus, the proposed adaptive CHT consists of self-tuning initial tolerance relaxations, and then decreasing these tolerances adaptively while using the PFPR technique to compare particles' performances within current tolerances. Note that the word *pseudo* is used in [33] because there are still a few parameters to be set. Nonetheless, the scheme may be viewed as adaptive by fixing these parameters to general-purpose values, as it has been done in this paper.

Bear in mind that only problems with side and inequality constraints are considered in this paper. The benefits of this technique are more evident when equality constraints are also present [33].

5.3. Termination Conditions

Termination conditions are important for a general-purpose optimizer because they allow setting a high $t_{\max}$ without resulting in unnecessarily large search-lengths for simple problems. Due to its population-based nature, PSO does not lend itself to traditional termination conditions used for single-solution methods. We propose some measures that can be used to infer convergence or stagnation in PSO, dividing them in two groups:

1) *Clustering measures*: within a single time-step.
2) *Evolution measures*: between time-steps.

In addition, these measures can be computed in terms of the positions in the search-space (preceded by *pb_* for *position-based*) or in terms of the values of the conflict function (preceded by *cb_*, for *conflict-based*). They can

also be computed in terms of the information currently held by the particles or in terms of their memorized information. Here we use the latter.

Two *clustering* measures are offered in Eqs. (33) and (34) and four *evolution* measures are offered in Eqs. (35) to (38), where n is the number of dimensions; m is the swarm-size; $\mathbf{x}$ is a particle's position; **gb** is the global best position; **cg** is the centre of gravity of the swarm; $\bar{c}$ is the average conflict in the swarm; *cgb* is the conflict of the global best position in the swarm; $(x_{j\max} - x_{j\min})$ is the j^{th} feasible interval; t is the current time-step; and *tref* is a number of time-steps over which these measures are averaged to smooth their oscillations (see also [33]).

For further reading on Termination Conditions in PSO, refer to [63].

$$\text{pb\_me}^{(t)} = \frac{\sum_{i=t-tref+1}^{t} \sqrt{\sum_{j=1}^{n} \frac{\sum_{k=1}^{m} \left(x_{kj}^{(i)} - gb_j^{(i)}\right)^2}{m \cdot \left(x_{j\max} - x_{j\min}\right)^2}}}{tref \cdot \sqrt{n}} \tag{33}$$

$$\text{pb\_cge}^{(t)} = \frac{\sum_{i=t-tref+1}^{t} \sqrt{\sum_{j=1}^{n} \left(\frac{cg_j^{(i)} - gb_j^{(i)}}{x_{j\max} - x_{j\min}} \right)^2}}{tref \cdot \sqrt{n}} \tag{34}$$

$$\text{cb\_av}^{(t)} = \frac{\sum_{i=t-tref+1}^{t} \text{abs}\left(\bar{c}^{(i)} - \bar{c}^{(i-1)}\right)}{tref} \tag{35}$$

$$\text{cb\_b}^{(t)} = \frac{\left(cgb^{(t-tref)} - cgb^{(t)}\right)}{tref} \tag{36}$$

$$\text{pb\_cg}^{(t)} = \frac{\sum_{i=t-tref+1}^{t} \sqrt{\sum_{j=1}^{n} \left(\frac{cg_j^{(i)} - cg_j^{(i-1)}}{x_{j\max} - x_{j\min}} \right)^2}}{tref \cdot \sqrt{n}} \tag{37}$$

$$\text{pb\_gb}^{(t)} = \frac{\sum_{i=t-tref+1}^{t} \sqrt{\sum_{j=1}^{n} \left(\frac{gb_j^{(i)} - gb_j^{(i-1)}}{x_{j\max} - x_{j\min}} \right)^2}}{tref \cdot \sqrt{n}} \tag{38}$$

6. Proposed Integrated Tool (PIT)

At the beginning of section 5, we argued that nature-inspired global search methods would be appropriate to seek the optimum management scheme in the strongly multimodal waterflooding (WF) problem, and proposed the use of a PSO algorithm. The details of the algorithm were presented in detail in the remainder of the section.

In addition to their ability to cope with multimodal problems, another advantage of nature-inspired methods in general and of PSO in particular is that the functions involved do not need to be differentiable, continuous, or even explicit. All that is needed is a way to evaluate the relative goodness of the potential solutions. If this cannot be achieved by the evaluation of an analytical function, it may be achieved by means of computational simulations, approximators, response surfaces, or even experiments.

The functions involved in the WF reservoir problem formulation in section 2 cannot be evaluated analytically. Therefore, a computational simulation is required, for which we have a commercial High-Fidelity (HF) reservoir simulator available [11]. However, every HF simulation requires a high computational effort. For instance, one single simulation using an i7-3.4GHz processor takes over a minute for Reservoir 1 in section 7.1; over 2 minutes for Reservoir 2 in section 7.2; while it could take several hours and even days for more complex reservoirs. This presents a problem for a population-based method like PSO, which typically requires a high number of function calls. Therefore, we propose the use of a remarkably cheaper Kriging surrogate model in place of the HF simulator. This model is trained offline using a much smaller number of simulations than would be required for the direct simulation-based optimization.

Thus, the Proposed Integrated Tool (PIT) consists of the integration of a commercial HF simulator; a Kriging surrogate model; a DoE technique to train the model offline; and a general-purpose PSO algorithm coupled with an adaptive CHT and termination conditions. A high-level description of the PIT is offered in Fig. 4, where the green text boxes describe modelling modules whereas the red text boxes describe solver modules. The flowchart in Fig. 4 may be described as follows:

1. Generate a number of sampling points using LCVT DoE technique.
2. Perform High Fidelity (HF) reservoir simulations on sampling points (IMEX).
3. Train Kriging Model (KM) using responses obtained from HF simulations.
4. Search for global optimum of static KM with Particle Swarm Algorithm (PSA) coupled with adaptive CHT (no additional HF simulation).
5. Validate results by evaluating trained KM on the coordinates of the best known solution, and also by performing an additional HF simulation on the coordinates of the best result returned by the PSA.

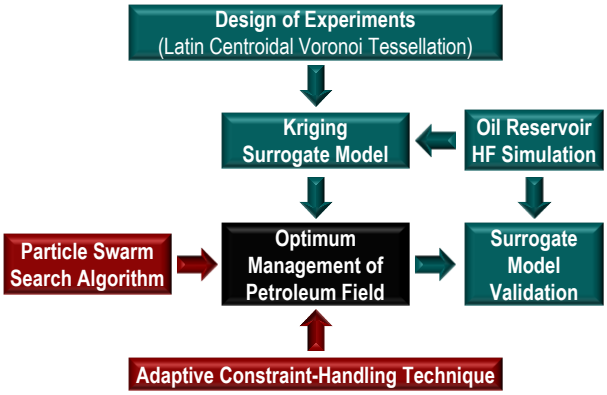

Fig. 4. High-level description of the Proposed Integrated Tool (PIT), where the green text boxes are associated with the modelling of the physical phenomena and the red text boxes are the techniques used to optimize the model.

7. Computational Experiments

The PIT is applied hereafter to two oil reservoir case studies. The features and settings of the PSA used are as described below, which are supported by experiments carried out on sets of benchmark problems in [33]:

- The termination conditions are, either the maximum number of time-steps permitted for the search (t_{max}) is reached, or the minimum number of time-steps permitted (t_{min}) is reached and the measures of clustering and evolution reach a maximum value that allows inferring convergence or stagnation. Refer to section 5.3 and Eqs. (33) to (38) for more details. The settings are as follows:
  1. Maximum search length $t_{max} = 10000$.
  2. Minimum search length $t_{min} = 500$.
  3. Termination values for clustering measures: $pb\_me = pb\_cge = 10^{-3}$.
  4. Termination values for evolution measures: $cb\_av = cb\_b = 10^{-6}$; $pb\_cg = pb\_gb = 10^{-3}$.
  5. $tref = 10$.
- A swarm of 50 particles is used, split in three sub-swarms. The first sub-swarm is composed of 17 particles governed by PSO-RRR2 formulation with $\omega = 0.8167$; the second one is composed of 16 particles governed by PSO-RRR1 formulation with $\omega = 0.80$; and the third one is composed of 17 particles governed by COPSO formulation with settings equivalent to $\phi_{min} = 0$ and $(\omega,\phi_{max}) = (0.7298,2.9922)$. For all sub-swarms, $\iota = 0.50$. Refer to section 5.1 for further details.
- An independent *forward topology* is implemented for each sub-swarm, sharing information among them by means of an *individual overlapping*. The number of neighbours in each sub-swarm is time-increasing linearly from 1 at the first time-step until it becomes global at $t = t_{max}$. Refer to section 5.1.3 for further details.
- Particles' positions are initialized by generating 1000 independent Latin Hypercube Samplings and selecting the one with the maximum minimum distance between particles. Each sub-swarm is initialized independently, and particles are initialized from stagnation. Every best experience (**pb**) is initialized at the same distance from its corresponding particle (**x**). Each component of this distance is calculated as the corresponding feasible interval divided by twice the number of particles in the swarm. The sign of the component and hence the direction of the distance vector are randomly generated. For each (**x**,**pb**) pair, a comparison is performed so that the best one becomes or stays **pb** and the other one becomes or stays **x** before the search begins. Thus, every particle starts the search with the same, moderate acceleration towards its best individual experience (**pb**).
- 25 runs are performed for each problem.
- Constraints are handled by the PFPR technique coupled with the adaptive scheme proposed in section 5.2, where *target FR* $\in$ [0.2,0.25]. For problems with FR > 0.25, *target FR* $\in$ $[(1.1 \cdot FR),(1.1 \cdot FR + 0.05)]$, with the obvious limit of 1. For problems with side constraints only, the latter are handled by the plain PFPR technique.

Note that no tuning is carried out in this paper. While even better results on some problems are likely possible for problem-specific parameter-tuning, the aim behind choosing a PSA with general-purpose settings, sub-swarms with complementary behaviour, and an adaptive CHT is to unburden the user of the hassle of performing costly and complex numerical trial-and-error testing and tuning on a case-by-case basis. Instead, the optimizer is run with settings that may be viewed as default. Thus, all problems hereafter are optimized using the same settings. The ultimate goal is to design a fully adaptive tool. Meanwhile, our PIT (see Fig. 4) trains the KM offline, uses general-purpose settings for the Search Algorithm and an adaptive CHT.

Furthermore, note that this PSA, with these settings, has also been tested on several benchmarking problems in [33], including the 5 unconstrained functions in [38], each with 2, 10 and 30 dimensions; the 13 constrained functions in [53]; the test suite of engineering problems in [77]; and the well-known 10-bar plane truss and 25-bar space truss design problems.

We also include results obtained optimizing the same KMs by the proposed PSA but without tolerance relation as well as by 3 commercial optimizers from the Matlab 2014a Optimization and Global Optimization Toolboxes, namely the local optimizer SQP and the global optimizers GlobalSearch (using Multi-Start SQP) and Genetic Algorithm (GA). While the aim is not to compare optimizers' performances, these values can be used as frames of reference to assess the benefits of the tolerance relaxation, the benefits of a global search, and also the accuracy of the proposed PSA with adaptive CHT irrespective of the accuracy of the KMs. In our experiments, the initial guess for SQP and Multi-Start SQP is in the middle of the bounding box defined by the feasible intervals, while the Multi-Start SQP considers 1000 potential initial guesses. For the GA, the population-size is set to 200 and $t_{max} = 2500$ so that the maximum number of objective Function Evaluations (FEs) permitted is the same one used for the proposed optimizer. All remaining settings are kept as default.

Bear in mind that *t* stands for *time-step* in the *optimization* context whereas it stands for *control cycle* in the *reservoir simulation* context.

7.1. Reservoir 1 (small, with 3 permeability zones)

The first reservoir has 1 injection and 2 production wells, as illustrated in Fig. 5 (from [78]). The reservoir has an area of 510x510 m^2 with a thickness of 4 m, which is modelled with a mesh of 51x51x1. The main characteristics of the reservoir are given in Table 2.

The arrangement of the injection and production wells and their regions are defined according to the horizontal permeability in the reservoir, as can be observed in Fig. 5. The horizontal permeability (k_h) in the injection well (I-1) region equals 1000 mD; k_h near the production well P-1 equals 500 mD; whereas k_h near the production well P-2 is equal to 1500 mD.

For this problem, $Q_{l,max}$ equals 40 m^3/day, and the individual flow cannot exceed 30 m^3/day. The $Q_{inj,max}$ equals 44 m^3/day. The rate at the injector is kept fixed whereas the rates at producers are variable during the optimization process. This leads to the following side constraints at producers (*P*) and at each control cycle (*t*): $0.25 \leq x_{p,t} \leq 0.75$. The total concession period considered is 16 years, and the oil price adopted is \$25.00/m^3. Other values used in the NPV calculations are: discount rate *d*

= 0.09; cost of water injection = \$2.00/m^3; and cost of water production = \$5.00/m^3.

As discussed in section 2 (see also Table 1), 4 cases are analyzed here according to the operational conditions and types of design variables:

- *Case 1*: FCO-FT considering different numbers of control cycles (side constrained problem).
- *Case 2*: NFCO-FT considering 3 control cycles.
- *Case 3*: FCO-VT considering 5 control cycles.
- *Case 4*: NFCO-VT considering 3 control cycles.

According to [2], inequality constraints are present in this problem for NFCO and/or VT operational conditions. The NFCO condition introduces $(3 \cdot n_t)$ inequalities whereas the VT condition introduces 1 inequality. Some statistics of the NPVs returned by our Proposed Integrated Tool (PIT) for each of these cases and sub-cases are shown in Table 3, Table 4 and Table 5.

Table 3 also shows the mean search length, the self-tuned initial relaxation of the tolerance and the resulting estimated FR within that tolerance, as well as problems' features such as the estimated FR, dimensionality (*n*), number of inequality (*ni*) and number of equality (*ne*) constraints. FRs are estimated by randomly generating 10^6 sampling points, checking their feasibility, and computing the ratio of the number of feasible over the number of infeasible generated points.

Table 4 also shows the best known solution for each problem using a Sequential Approximate Optimization (SAO) procedure equipped with a SQP local optimizer and a Trust Region (TR) based method for the update of the search-space for each local solution [79]. It also includes the results of the High-Fidelity (HF) evaluations of the coordinates of the best results returned by our PIT, and the results of the evaluation of the KMs on the coordinates of the best known solutions. These last two pieces of information are helpful to evaluate the accuracy of the KMs in the vicinity of the best known solutions and in the vicinity of the best results found by the proposed optimizer. A comparison between the best known solutions and the results returned by our PIT enables the assessment of its accuracy as a whole.

Table 5 also shows statistics of NPVs returned by our proposed PSA but without tolerance relation as well as by local optimizer SQP and global optimizers Multi-Start SQP and GA on the same KMs.

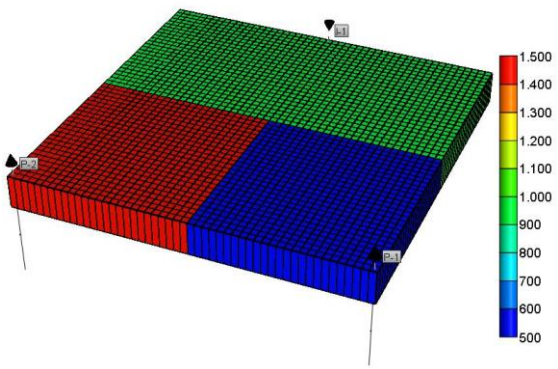

Fig. 5. Reservoir 1: Permeability field and well locations.

Table 2
Characteristics of Reservoir 1.

Simulation mesh	51x51x1 (510x510x4 m^3)
Porosity	30%
Horizontal permeability (k_h)	500 to 1500 mD
Vertical permeability (k_v)	10% of k_h
Rock compressibility at 200 kgf/cm²	5 x 10^{-5} $(kgf/cm^2)^{-1}$
Viscosity	0.97 cP
Gas-Oil Ratio (GOR)	115.5 m^3/m^3 std
Maximum platform liquid production rate	40 m^3/day
Maximum platform injection rate	44 m^3/day
Maximum liquid rate at producers	30 m^3/day
Minimum liquid rate at producers	10 m^3/day

7.1.1. Optimization of Reservoir 1 for Case 1

Three problems with side constraints only and different dimensionalities are considered here, where the number of control cycles equals the number of design variables (*n*). In order to observe the influence on the NPV response, $n = 2$, $n = 12$ and $n = 24$ design variables are considered. Each design variable represents the rates at producer *P1* at one production cycle.

The first step is to train the KM, for which 10 sampling points are used per design variable. Thus, 20, 120 and 240 HF simulations are performed, respectively.

This problem with $n = 2$, $n = 12$, and $n = 24$ is referred to as *Problem 1*, *Problem 5*, and *Problem 6*, respectively, in Table 3, Table 4 and Table 5. The control cycles are shown in Fig. 6.

These problems have side constraints only, and not even dimensionality seems to pose any difficulty for our PSA. As shown in Table 3 and Table 4, every run converged to the same result, and did so very quickly. In fact, the termination condition $t \geq t_{min}$ prevented the search from terminating before 500 time-steps have elapsed for *Problem 1* ($n = 2$) and for *Problem 5* ($n = 12$), whilst the search was terminated after 647 time-steps have elapsed, on average, for *Problem 6* ($n = 24$).

As shown in Table 5, all global search algorithms converged to this same NPV in all 25 runs, displaying very small sample standard deviations (smallest for Multi-Start SQP and largest for GA). Note that, since there are no constraints other than side constraints, our proposed PSA with and without tolerance relaxation are the same. Results also show that the very efficient local optimizer SQP falls short on these problems.

As shown in Table 4, the best known solutions of these problems are very similar to the best results obtained by our PIT, only differing in 0.01%, 0.15% and 0.07% of the best known solutions.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best results found

Table 3
Problems' features and statistics of NPVs returned by Proposed Integrated Tool (PIT) for Reservoir 1, scaled by a factor of 10^{-6}. Additional information such as average search length, tolerance relaxation, and Feasibility Ratios (FRs) is also offered, where FRs are estimated by checking the feasibility of 10^6 randomly generated sample points within side constraints.

PROBLEM		FR [%]	*n*	*ni*	*ne*	FR Relaxed Tol. [%]	Relaxed Initial Tol. Ineq.	NPVs returned by Proposed Integrated Tool (PIT) x 10^{-6}					Mean Final *t*
No.	Denomination							Best	Median	Mean	Worst	Sample σ	
1	2dv_FCO	100	2	0	0	-	-	**1.4924**	**1.4924**	**1.4924**	**1.4924**	**4.6000E-16**	500
2	9dv_FCOdt	79.8503	9	1	0	89.18	0.0927	**1.5167**	**1.5167**	**1.5167**	**1.5167**	**9.6002E-12**	2739
3	9dv_NFCO	0.0087	9	9	0	22.75	0.4215	**1.7151**	**1.7145**	**1.7137**	**1.6985**	**3.2619E-03**	10000
4	11dv_NFCOdt	0.008	11	10	0	22.84	0.4223	**1.7216**	**1.7216**	**1.7160**	**1.6521**	**1.9191E-02**	9847
5	12dv_FCO	100	12	0	0	-	-	**1.4957**	**1.4957**	**1.4957**	**1.4957**	**9.7139E-12**	500
6	24dv_FCO	100	24	0	0	-	-	**1.4938**	**1.4938**	**1.4938**	**1.4938**	**7.3492E-09**	647

Table 4
Statistics of NPVs returned by PIT for Reservoir 1, best known solutions from [79], High Fidelity (HF) evaluations on best coordinates returned by PIT, and Kriging Models evaluations on coordinates of best known solutions. Values are scaled by 10^{-6}.

PROBLEM		Best Known Solution x 10^{-6}	NPVs returned by Proposed Integrated Tool (PIT) x 10^{-6}					HF evaluation on coordinates returned by PIT x 10^{-6}	KM evaluation on coordinates of Best Known Solution x 10^{-6}
No.	Denomination		Best	Median	Mean	Worst	Sample σ		
1	2dv_FCO	**1.4922**	**1.4924**	1.4924	1.4924	1.4924	4.6000E-16	**1.4920**	**1.4922**
2	9dv_FCOdt	**1.4936**	**1.5167**	1.5167	1.5167	1.5167	9.6002E-12	**1.4867**	**1.4823**
3	9dv_NFCO	**1.7240**	**1.7151**	1.7145	1.7137	1.6985	3.2619E-03	**1.5950**	**1.6190**
4	11dv_NFCOdt	**1.7259**	**1.7216**	1.7216	1.7160	1.6521	1.9191E-02	**1.7100**	**1.5020**
5	12dv_FCO	**1.4935**	**1.4957**	1.4957	1.4957	1.4957	9.7139E-12	**1.4923**	**1.4892**
6	24dv_FCO	**1.4928**	**1.4938**	1.4938	1.4938	1.4938	7.3492E-09	**1.4927**	**1.4909**

by our PIT, resulting in errors of our KMs of 0.03% of the HF simulation for *Problem 1* (n = 2); 0.23% for *Problem 5* (n = 12); and 0.07% for *Problem 6* (n = 24). Another accuracy check was carried out by evaluating the KMs on the coordinates of the best known solutions, resulting in errors of our KMs of 0.00% of the best known solution for *Problem 1* (n = 2); 0.29% for *Problem 5* (n = 12); and 0.13% for *Problem 6* (n = 24) (see Table 4).

7.1.2. Optimization of Reservoir 1 for Case 2

In order to study the impact of more flexible management schedules, we investigate in this section the case of a Non-Full Capacity Operation (NFCO) of the problem discussed in the previous section (see Table 1). The control cycles considered are shown in Fig. 7 with fixed switching times (FT). Since the rates at all three wells are variable, and the number of control cycles equals 3 (n_t = 3), this problem presents a total of 9 design variables ($n = 3 \cdot n_t = 9$) and 9 inequality constraints ($ni = 3 \cdot n_t = 9$) (see Table 3). This problem is referred to as *Problem 3* in Table 3, Table 4 and Table 5.

As in the previous cases, 10 sampling points are used per design variable to train the KM, leading to 90 HF simulations.

While there is no equality constraint, the problem is still highly constrained with an estimated FR = 0.0087%, as shown in Table 3. After the relaxation of the tolerance to 0.4215, the resulting estimated FR = 22.75%, which is within the range specified in the settings (20% to 25%).

This problem is noticeably harder than those in the previous section. Nevertheless, as shown in Table 3 and Table 4, our optimizer obtains good results consistently: the best NPV found in 25 runs equals 1.7151; the median NPV equals 1.7145; the mean NPV equals 1.7137; and the worst NPV equals 1.6985; with a sample standard deviation (σs) of 0.003. The maximum search-length permitted was reached in every run. Nevertheless, the mean convergence curve is reasonably flat by the end of the search, as shown in Fig. 8 (left).

Further analysis on the data extracted from the search (not presented for brevity) shows that the difference in coordinates between the best and the median results is minor in every dimension, and therefore they are in the same region of the search-space. In fact, the maximum and average absolute differences between corresponding coordinates of the best and of the median results returned equal 6.14% and 2.45% of the corresponding feasible interval, respectively. Hence it can be inferred that the difference is in fine-tuning. Only in 1 out of 25 runs is the result in a different region of the search-space.

As shown in Table 5, the Multi-Start SQP converged to the same NPV, namely 1.7151, in all 25 runs. Our PSA with and without tolerance relaxation is able to find

Table 5
Statistics of NPVs returned by proposed PSA with adaptive CHT on KMs for Reservoir 1. Statistics of NPVs returned by the same PSA but without tolerance relation as well as by 3 commercial optimizers, namely the local optimizer SQP and the global optimizers Multi-Start SQP and Genetic Algorithm (GA), are offered as frames of reference. Recall that values are scaled by 10^{-6}.

PROBLEM		Algorithm	NPVs returned by different Optimizers on our Kriging Models x 10^{-6}					Mean Obj. FEs	No. runs
No.	Denomination		Best	Median	Mean	Worst	Sample σ		
1	2dv_FCO	**Proposed PSA + Adaptive CHT**	**1.4924**	**1.4924**	**1.4924**	**1.4924**	**4.6000E-16**	24985	25
		PSA without Tolerance Relaxation	1.4924	1.4924	1.4924	1.4924	4.6000E-16	24985	25
		Matlab 2014a SQP	1.4900	-	-	-	-	38	1
		Matlab 2014a Multi-Start SQP	1.4924	1.4924	1.4924	1.4924	4.5325E-16	1040	25
		Matlab 2014a Genetic Algorithm	1.4924	1.4924	1.4924	1.4924	7.5150E-13	53211	25
2	9dv_FCOdt	**Proposed PSA + Adaptive CHT**	**1.5167**	**1.5167**	**1.5167**	**1.5167**	**9.6002E-12**	109184	25
		PSA without Tolerance Relaxation	1.5167	1.5167	1.5164	1.5111	1.1200E-03	106570	25
		Matlab 2014a SQP	1.5111	-	-	-	-	260	1
		Matlab 2014a Multi-Start SQP	1.5167	1.5167	1.5167	1.5167	1.0336E-13	18276	25
		Matlab 2014a Genetic Algorithm	1.5167	1.5167	1.5155	1.5090	2.4832E-03	52754	25
3	9dv_NFCO	**Proposed PSA + Adaptive CHT**	**1.7151**	**1.7145**	**1.7137**	**1.6985**	**3.2619E-03**	162701	25
		PSA without Tolerance Relaxation	1.7151	1.7140	1.7122	1.6993	4.1722E-03	178718	25
		Matlab 2014a SQP	1.4805	-	-	-	-	86	1
		Matlab 2014a Multi-Start SQP	1.7151	1.7151	1.7151	1.7151	1.1808E-13	14581	25
		Matlab 2014a Genetic Algorithm	1.7134	1.6954	1.6916	1.6262	1.7018E-02	53883	25
4	11dv_NFCOdt	**Proposed PSA + Adaptive CHT**	**1.7216**	**1.7216**	**1.7160**	**1.6521**	**1.9191E-02**	168945	25
		PSA without Tolerance Relaxation	1.7216	1.7215	1.6907	1.6497	3.5523E-02	168512	25
		Matlab 2014a SQP	1.7216	-	-	-	-	229	1
		Matlab 2014a Multi-Start SQP	1.7216	1.7216	1.7216	1.7216	7.8983E-14	18376	25
		Matlab 2014a Genetic Algorithm	1.7214	1.6365	1.6390	1.5306	6.5535E-02	119448	25
5	12dv_FCO	**Proposed PSA + Adaptive CHT**	**1.4957**	**1.4957**	**1.4957**	**1.4957**	**9.7139E-12**	20584	25
		PSA without Tolerance Relaxation	1.4957	1.4957	1.4957	1.4957	9.7139E-12	20584	25
		Matlab 2014a SQP	1.4943	-	-	-	-	860	1
		Matlab 2014a Multi-Start SQP	1.4957	1.4957	1.4957	1.4957	6.7987E-16	1794	25
		Matlab 2014a Genetic Algorithm	1.4957	1.4957	1.4957	1.4957	1.1476E-08	53211	25
6	24dv_FCO	**Proposed PSA + Adaptive CHT**	**1.4938**	**1.4938**	**1.4938**	**1.4938**	**7.3492E-09**	30787	25
		PSA without Tolerance Relaxation	1.4938	1.4938	1.4938	1.4938	7.3492E-09	30787	25
		Matlab 2014a SQP	1.4934	-	-	-	-	2380	1
		Matlab 2014a Multi-Start SQP	1.4938	1.4938	1.4938	1.4938	4.5325E-16	3101	25
		Matlab 2014a Genetic Algorithm	1.4938	1.4938	1.4938	1.4938	3.5587E-06	53211	25

this NPV in some but not all runs, where the one with the tolerance relaxation exhibits better performance in terms of median, mean and σs. In turn, the GA exhibits worse performance whereas the result returned but the efficient SQP is very poor. Thus, results in Table 5 show that our proposed optimizer obtains good results, being the second best among the ones compared on this KM. They also show that a global search is necessary, and that the tolerance relaxation is beneficial.

As shown in Table 4, the best known solution of this problem (1.7240) is very similar to the best result returned by our PIT (1.7151), only differing in 0.52% of the best known solution. Comparing this (NFCO-FT) to the FCO-FT operational conditions, it can be concluded that the former improves the NPV with fewer control cycles and fewer design variables.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best result found by our PIT, resulting in an error of our KM of 7.53% of the HF simulation. Another accuracy check was carried out by evaluating the KM on the coordinates of the best known solution, resulting in an error of our KM of 6.09% of the best known solution (see Table 4).

7.1.3. Optimization of Reservoir 1 for Case 3

In this problem, the switching times for the control cycles are included as design variables for the FCO condition (FCO-VT in Table 1). 5 control cycles are chosen, leading to a problem with 9 design variables (n = 9) and 1 inequality constraint (ni = 1), the latter due to the VT condition (see Table 3). This problem is referred to as *Problem 2* in Table 3, Table 4 and Table 5.

As before, 10 sampling points are used per design variable to train the KM, leading to 90 HF simulations.

This problem presents a high FR = 79.85%, as shown in Table 3. While this is already notably higher than the target FR of 20% to 25%, the automatic self-tuning mechanism of the CHT relaxes the tolerance to 0.0927, which results in an estimated FR = 89.18%. This relaxation is especially useful in cases where the solution lies on or near the boundaries of the feasible region.

For this problem, our optimizer obtains the same NPV in every run (1.5167), for which it requires 2739 time-steps on average (see Table 3 and Table 4).

As shown in Table 5, our proposed optimizer and the Multi-Start SQP converged to the same NPV, namely 1.5167, in all 25 runs. The best and the median of the NPVs obtained by the PSA without tolerance relaxation and also by the GA are also equal to 1.5167, but they fail to achieve this value in some runs. Among these last two algorithms, GA exhibits marginally worse performance in terms of the mean and σs. Once again, the NPV returned but the efficient SQP is the worst one. Thus, results in Table 5 show that our proposed optimizer and the Multi-Start SQP achieve the best results on this KM. They also show that a global search is necessary, and that the tolerance relaxation is beneficial here as well.

As shown in Table 4, the best known solution of this problem (1.4936) is similar to the best result returned by our PIT (1.5167), only differing in 1.55% of the best known solution. Comparing this solution (FCO-VT) to those obtained with FCO-FT, it can be concluded that this operational condition marginally improves the NPV.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best result found by our PIT, resulting in an error of our KM of 2.02% of the HF simulation. Another accuracy check was carried out by evaluating the KM on the coordinates of the best known solution, resulting in an error of our KM of 0.76% of the best known solution (see Table 4).

7.1.4. Optimization of Reservoir 1 for Case 4

This is the most generic formulation defined in section 2 (see Table 1), resulting in the most flexible reservoir management process. Thus, 3 control cycles and NFCO-VT operational condition are chosen, leading to a problem with 11 design variables (n = 11) and 10 inequality constraints (ni = 10). As in *Case 2* in section 7.1.2, the number of inequality constraints due to NFCO with 3 control cycles equals 9, whilst the VT condition introduces 1 additional inequality constraint as in *Case 3* in section 7.1.3. This problem is referred to as *Problem 4* in Table 3, Table 4 and Table 5.

As before, 10 sampling points are used per design variable to train the KM, leading to 110 HF simulations.

While there is no equality constraint, this problem is still highly constrained with an estimated FR = 0.008%, as shown in Table 3. After the relaxation of the tolerance to 0.4223, the resulting estimated FR = 22.84%, which is within the range specified in the settings (20% to 25%).

For this problem, our optimizer is able to obtain very good results consistently (see Table 3 and Table 4), converging to the best NPV in 23 out of 25 runs. Thus, both the best and the median NPVs equal 1.7216; the mean NPV equals 1.7160; and the worst NPV equals 1.6521; with σs = 0.019 and an average search-length of 9847 time-steps. All 10000 time-steps permitted for the search were used up in 24 out of 25 runs. Nevertheless, the mean convergence curve is reasonably flat by the end of the search, as shown in Fig. 8 (right).

As shown in Table 5, the Multi-Start SQP converged to the same NPV, namely 1.7216, in all 25 runs. Our

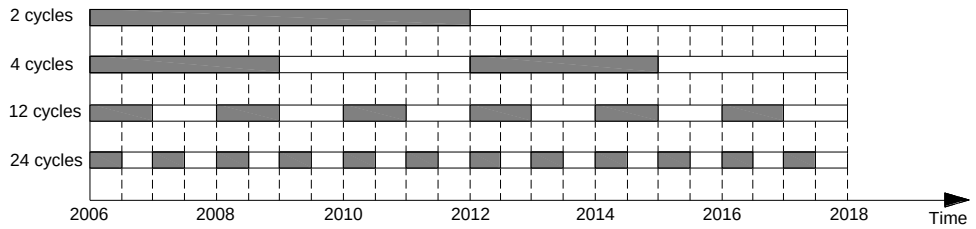

Fig. 6. Control cycles for FCO-FT operational conditions (Reservoir 1; Case 1; Problems 1, 5 and 6).

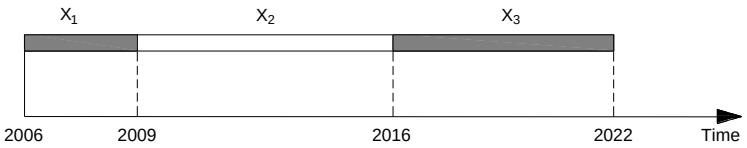

Fig. 7. Control cycles for NFCO-FT operational condition (Reservoir 1, Case 2, Problem 3).

proposed optimizer converged to the same NPV in 23 out of 25 runs. In turn, the PSA without tolerance relaxation obtained good results, though not quite as good, whereas the GA was able to achieve considerable worse performance in this case. Strangely enough, the result returned but the efficient SQP is spot on this time. Thus, results in Table 5 show that our proposed optimizer obtains very good results on this KM, almost as good as those obtained by the Multi-Start SQP and by the efficient plain SQP. They also show that the tolerance relaxation is beneficial.

As shown in Table 4, the best known solution of this problem (1.7259) is very similar to the best result returned by our PIT (1.7216), only differing in 0.25% of the best known solution. Comparing these results against those obtained with the other operational conditions, this operational condition results in the best optimized NPV, which is achieved considering 3 control cycles only.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best result found by our PIT, resulting in an error of our KM of 0.68% of the HF simulation. Another accuracy check was carried out by evaluating the KM on the coordinates of the best known solution, resulting in an error of our KM of 12.97% of the best known solution (see Table 4).

7.2. Reservoir 2 (with real permeability field)

The second reservoir investigated here is the more realistic, medium-sized reservoir shown in Fig. 9. This case was taken from [78], and is a synthetic form based on the Brush Canyon Outcrop data [80]. It has 12 wells, 7 producers and 5 injectors. The numerical model consists of a grid of 43x55x6 blocks, whose main characteristics are shown in Table 6. According to [2], the FCO condition in this reservoir results in $4 \cdot n_t$ inequality constraints ($ni = 4 \cdot n_t$). The same as in Reservoir 1, the NFCO condition introduces $(3 \cdot n_t)$ inequalities, whilst the VT condition introduces 1 more.

Thus, three problems are analyzed hereafter with 2 operational conditions: FCO-FT and NFCO-VT. Some statistics of the NPVs returned by our PIT for these problems are shown in Table 7, Table 8 and Table 9.

Table 7 also shows the mean search length, the self-tuned initial relaxation of the tolerance and the resulting estimated FR within that tolerance, as well as problems' features such as the estimated FR, dimensionality (n), number of inequality (ni) and number of equality (ne) constraints. FRs are estimated by randomly generating 10^6 sampling points, checking their feasibility, and computing the ratio of the number of feasible over the number of infeasible generated points.

Table 8 also shows the best known solution for each problem using a Sequential Approximate Optimization (SAO) procedure equipped with a SQP local optimizer and a Trust Region (TR) based method for the update of the search-space for each local solution [79]. It also includes the results of the HF evaluations of the

Table 6
Characteristics of Reservoir 2.

Simulation mesh	43x55x6
Porosity	16% to 28%
Horizontal permeability (k_h)	157 to 2592 mD
Vertical permeability (k_v)	30% of k_h
Rock compressibility at 1019 kgf/cm²	2 x 10^{-7} (kgf/cm²)$^{-1}$
Viscosity	0.11 cP
Gas-Oil Ratio (GOR)	78.1 m^3/m^3std
Maximum platform liquid production rate	5000 m^3/day
Maximum platform injection rate	5500 m^3/day
Maximum liquid rate at producers	900 m^3/day
Maximum water rate at injectors	1500 m^3/day

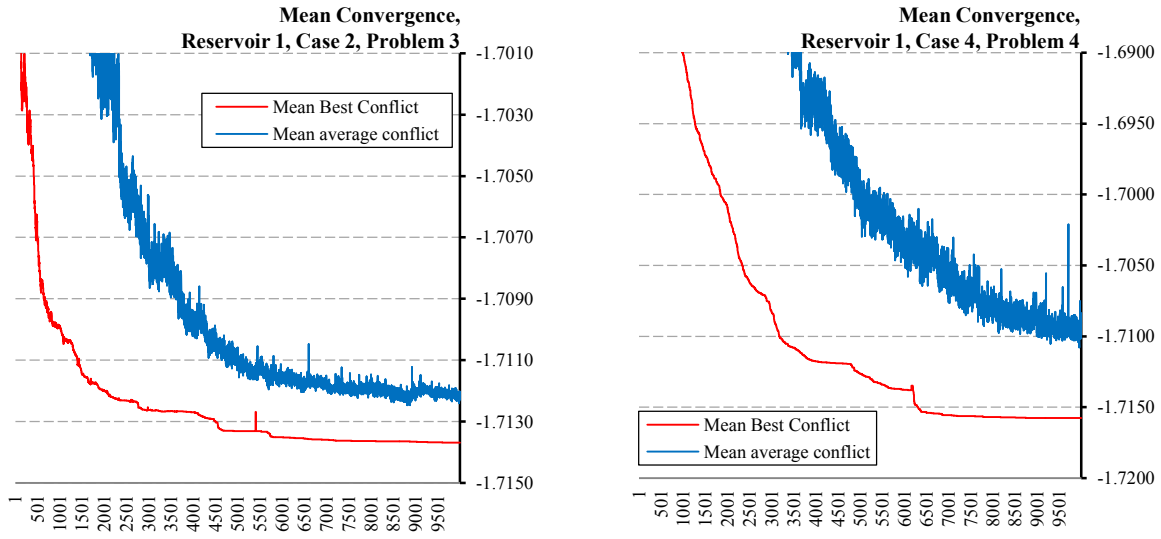

Fig. 8. Mean convergence curves for Problems 3 and 4, where conflict stands for –NPVx10^{-6}, average is calculated among all feasible particles in a given time-step of a given run, and mean is calculated among 25 runs.

coordinates of the best results returned by our PIT, and the results of the evaluation of the KMs on the coordinates of the best known solutions. These last two pieces of information are helpful to evaluate the accuracy of the KMs in the vicinity of the best known solutions and in the vicinity of the best results found by the proposed optimizer. A comparison between the best known solutions and the results returned by our PIT enables the assessment of its accuracy as a whole.

Table 9 also shows statistics of NPVs returned by our proposed PSA but without tolerance relation as well as by local optimizer SQP and global optimizers Multi-Start SQP and GA on the same KMs.

7.2.1. Optimization of Reservoir 2 for FCO-FT Case

This case is analyzed with 1 and 6 control cycles, as shown in Fig. 10. This leads to a problem with 10 design variables (n = 10) and 4 inequality constraints (ni = 4) in the former case, and to a problem with 60 design variables (n = 60) and 24 inequality constraints (ni = 24) in the latter case. In Table 7, Table 8 and Table 9, these problems are referred to as *Problem 7* for 1 control cycle, and as *Problem 9* for 6 control cycles.

Problem with 1 Control Cycle (Problem 7)

For this problem, 10 sampling points are used per design variable, thus performing 100 HF simulations for the training of the KM.

While there is no equality constraint, this problem is still highly constrained with an estimated FR = 0.0794%, as shown in Table 7. After the relaxation of the tolerance to 0.2532, the resulting estimated FR = 23.65%, which is within the range specified in the settings (20% to 25%).

For this problem, our optimizer obtains very good results consistently (see Table 7 and Table 8): the best NPV found in 25 runs equals 300.5358; the median NPV equals 300.5345; the mean NPV equals 300.5341; and the worst NPV equals 300.5305; with a σs = 0.0014. The maximum search length permitted of 10000 time-steps was reached in every run. Nevertheless, the convergence curve in Fig. 11 (left) shows that practical convergence has been achieved. CVs are calculated disregarding tolerance relaxation and therefore are actual violations.

Further analysis on the data extracted from the search (not presented for brevity) shows that the difference in coordinates between the best and the worst results is minor in every dimension, and therefore they are in the same region of the search-space. In fact, the maximum and average absolute differences between corresponding coordinates of the best and of the worst results returned

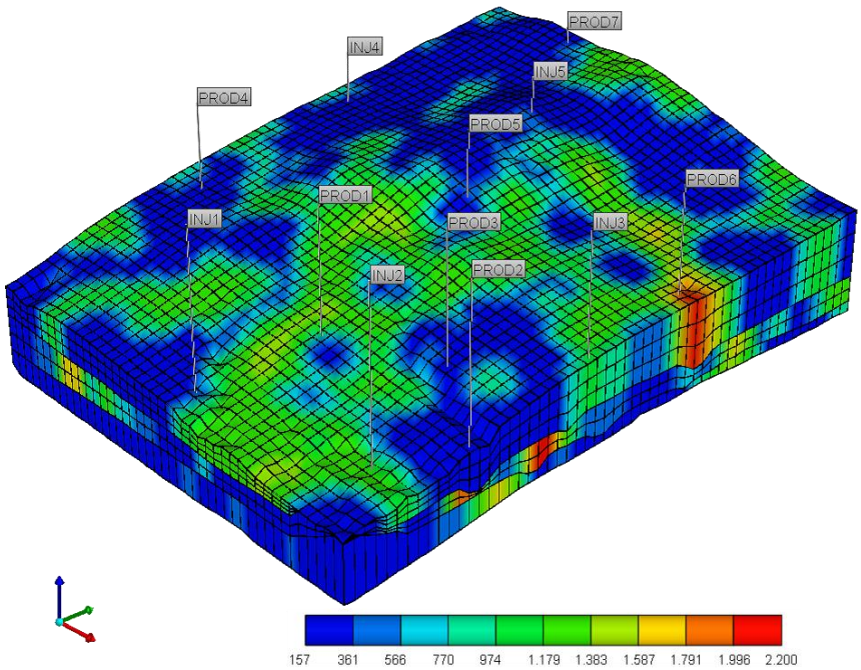

Fig. 9. Reservoir 2: Permeability field and well locations.

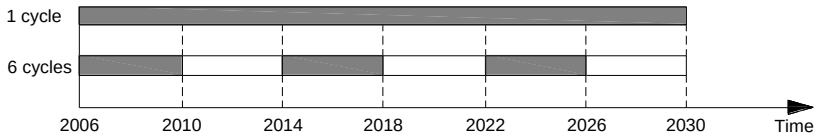

Fig. 10. Control cycles for FCO-FT operational condition (Reservoir 2, Case 1, Problems 7 and 9).

Table 7
Problems' features and statistics of NPVs returned by Proposed Integrated Tool (PIT) for Reservoir 2, scaled by a factor of 10^{-6}. Additional information such as average search length, tolerance relaxation, and Feasibility Ratios (FRs) is also offered, where FRs are estimated by checking the feasibility of 10^6 randomly generated sample points within side constraints.

PROBLEM		FR [%]	*n*	*ni*	*ne*	FR Relaxed Tol. [%]	Relaxed Initial Tol. Ineq.	NPVs returned by Proposed Integrated Tool (PIT) x 10^{-6}					Mean Final *t*
No.	Denomination							Best	Median	Mean	Worst	σs	
7	10dv_FCO	0.0794	10	4	0	23.65	0.2532	**300.5358**	**300.5345**	**300.5341**	**300.5305**	**0.0014**	10000
8	38dv_NFCOdt	0.3391	38	10	0	24.29	0.1799	**290.5559**	**263.6527**	**267.3696**	**241.2921**	**13.1455**	9666
9	60dv_FCO	$< 10^{-4}$	60	24	0	24.14	0.4253	**282.7148**	**252.9083**	**258.5174**	**252.3202**	**10.8881**	10000

Table 8
Statistics of NPVs returned by PIT for Reservoir 2, best known solutions from [79], High Fidelity (HF) evaluations on best coordinates returned by PIT, and Kriging Models evaluations on coordinates of best known solutions. Values are scaled by 10^{-6}.

PROBLEM		Best Known Solution x 10^{-6}	NPVs returned by Proposed Integrated Tool (PIT) x 10^{-6}					HF evaluation on coordinates returned by PIT x 10^{-6}	KM evaluation on coordinates of Best Known Sol. x 10^{-6}
No.	Denomination		Best	Median	Mean	Worst	σs		
7	10dv_FCO	**303.8614**	**300.5358**	300.5345	300.5341	300.5305	0.0014	**285.9695**	**265.6160**
8	38dv_NFCOdt	**346.3539**	**290.5559**	263.6527	267.3696	241.2921	13.1455	**265.5500**	**204.8254**
9	60dv_FCO	**287.5235**	**282.7148**	252.9083	258.5174	252.3202	10.8881	**279.8260**	**252.1032**

equal 0.75% and 0.22% of the corresponding feasible interval, respectively. Hence it can be inferred that the difference is in fine-tuning.

As shown in Table 9, the Multi-Start SQP converged to the same NPV, namely 300.5359, in all 25 runs. Our proposed optimizer converges around the same solution in every run, despite not being able to complete the fine-tuning of the search; note that the search is terminated due to reaching t_{max} in every run. In turn, the PSA without tolerance relaxation obtained marginally worse results, whereas the GA displays considerable worse performance in this case. Strangely enough, the result returned but the efficient SQP is spot on this time. Thus, results in Table 9 show that our proposed optimizer obtains very good results on this KM, almost as good as those obtained by the Multi-Start SQP and by the efficient plain SQP. They also show that the tolerance relaxation is beneficial.

As shown in Table 8, the best known solution of this problem (303.8614) is similar to the best result returned by our PIT (300.5358), only differing in 1.09% of the best known solution.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best result found by our PIT, resulting in an error of our KM of 5.09% of the HF simulation. Another accuracy check was carried out by evaluating the KM on the coordinates of the best known solution, resulting in an error of our KM of 12.59% of the best known solution (see Table 8).

Problem with 6 Control Cycles (Problem 9)

For this problem, only 6 sampling points are used per design variable, thus performing 360 HF simulations for the training of the KM.

While there is no equality constraint, this problem is still highly constrained with FR < 10^{-4}%, as shown in Table 7. After the relaxation of the tolerance to 0.4253, the resulting estimated FR = 24.14%, which is within the range specified in the settings (20% to 25%).

Search Algorithms in general, and PSO in particular, suffer from the *curse of dimensionality*. This means that the performance of the method quickly deteriorates as the dimensionality of the search-space increases. The PSA used here is a general-purpose one, equipped with no additional strategy to cope with large-scale problems.

Nonetheless, our optimizer is able to obtain good results, although not as consistently as before (see Table 7 and Table 8): the best NPV found in 25 runs equals 282.7148; the median NPV equals 252.9083; the mean NPV equals 258.5174; and the worst NPV equals 252.3202; with a σs = 10.8881. The search went through all 10000 time-steps permitted in every run.

As shown in Fig. 11 (right), the extremely low FR has a strong influence on the dynamics of the search, as compliance with all constraints proves remarkably more difficult than in all previous problems. The adaptive update of the relaxed tolerance is not sufficient to reach zero tolerance by the time 72% of t_{max} has elapsed. Therefore, additional updates are enforced to ensure that it is reached by the time 80% of t_{max} has elapsed. Compare this with the tolerance updates for *Problem 7* in Fig. 11 (left), which reaches the zero tolerance before 250 time-steps have elapsed. It is also interesting to note that the CV of the best candidate solution in the swarm is consistently higher than the average CV of the swarm. Also note that, while the tolerance is relaxed, the curves of not only the best but also of the average conflict fall below the best feasible candidate solution finally found, which suggests that solutions are being pushed by

Table 9
Statistics of NPVs returned by proposed PSA with adaptive CHT on KMs for Reservoir 2. Statistics of NPVs returned by the same PSA but without tolerance relation as well as by 3 commercial optimizers, namely the local optimizer SQP and the global optimizers Multi-Start SQP and Genetic Algorithm (GA), are offered as frames of reference. Recall that values are scaled by 10^{-6}.

PROBLEM		Algorithm	NPVs returned by different Optimizers on our Kriging Models x 10^{-6}					Mean Obj. FEs	No. runs
No.	Denomination		Best	Median	Mean	Worst	Sample σ		
7	10dv_FCO	**Proposed PSA + Adaptive CHT**	**300.5358**	**300.5345**	**300.5341**	**300.5305**	**1.4177E-03**	184087	25
		PSA without Tolerance Relaxation	300.5357	300.5340	300.5328	300.5242	3.0933E-03	182340	25
		Matlab 2014a SQP	300.5359	-	-	-	-	322	1
		Matlab 2014a Multi-Start SQP	300.5359	300.5359	300.5359	300.5359	1.6176E-11	20993	25
		Matlab 2014a Genetic Algorithm	300.4864	300.2907	300.2119	299.5569	2.3146E-01	25987	25
8	38dv_NFCOdt	**Proposed PSA + Adaptive CHT**	**290.5559**	**263.6527**	**267.3696**	**241.2921**	**1.3146E+01**	171422	25
		PSA without Tolerance Relaxation	290.5543	261.7965	264.2924	231.0983	1.7895E+01	172653	25
		Matlab 2014a SQP	280.0735	-	-	-	-	3831	1
		Matlab 2014a Multi-Start SQP	290.5853	286.8488	283.8809	261.0225	6.6127E+00	151555	25
		Matlab 2014a Genetic Algorithm	286.2416	260.5684	257.8816	229.1166	1.5151E+01	63622	25
9	60dv_FCO	**Proposed PSA + Adaptive CHT**	**282.7148**	**252.9083**	**258.5174**	**252.3202**	**1.0888E+01**	121558	25
		PSA without Tolerance Relaxation	282.7140	254.7223	261.7703	252.2277	1.3197E+01	150977	25
		Matlab 2014a SQP	254.6023	-	-	-	-	4559	1
		Matlab 2014a Multi-Start SQP	282.7150	282.7150	282.7150	282.7150	6.9994E-11	358559	25
		Matlab 2014a Genetic Algorithm	282.7002	252.2455	261.5048	252.2265	1.3428E+01	46641	25

tolerance updates towards the progressively smaller feasible space. For *Problem 7*, the curve of the best conflict falls below the best feasible candidate solution finally found only at the early stages of the search, whereas the average conflict is always above.

Further analysis on the data extracted from the search (not presented for brevity) shows that only in 4 out of 25 runs did the results converge to the same region of the search-space where the best result was found. The remaining 21 runs return results in different regions of the search-space, not close to one another, all displaying conflicts in the range 252 to 256.

As shown in Table 9, the Multi-Start SQP converged to the same NPV, namely 282.7150, in all 25 runs. Our proposed optimizer converges around the same result (282.7) in 4 out of 25 runs, converging to results in the range of 253 to 256 for the remaining 21 runs. In turn, the PSA without tolerance relaxation found marginally better results, converging around 282.7 in 6 out of 25 runs, and around results in the range of 252 to 257 for the remaining 19 runs. The GA exhibit similar performance in this case, whilst the result returned by SQP is similar to the median NPV returned by our PSA with and without tolerance relaxation and by the GA. Thus, results in Table 9 show that our proposed optimizer obtains results on this KM which are worse than those obtained by the Multi-Start SQP and competitive with the other Global Search algorithms. It seems that the tolerance relaxation does not have a strong beneficial effect if the adaptive scheme does not manage to eliminate it in time, and therefore the search is mostly driven by tolerance updates pushing the best candidate solutions towards the progressively smaller feasible space. This issue needs to be studied further.

As shown in Table 8, the best known solution of this problem (287.5235) is similar to the best result returned by our PIT (282.7148), only differing in 1.67% of the best known solution.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best result found by our PIT, resulting in an error of our KM of 1.03% of the HF simulation. Another accuracy check was carried out by evaluating the KM on the coordinates of the best known solution, resulting in an error of our KM of 12.32% of the best known solution (see Table 8).

7.2.2. Optimization of Reservoir 2 for NFCO-VT Case

This case is analyzed with 3 control cycles (n_t = 3) and NFCO operational condition, leading to a problem with 38 design variables (n = 38) and 10 inequality constraints ($ni = 3 \cdot n_t + 1 = 10$). In Table 7, Table 8 and Table 9, this problem is referred to as *Problem 8*.

For this problem, 10 sampling points are used per design variable, leading to 380 HF simulations to train the KM.

While there is no equality constraint, this problem is still highly constrained with estimated FR = 0.3391%, as shown in Table 7. After the relaxation of the tolerance to 0.1799, the resulting estimated FR = 24.29%, which is within the range specified in the settings (20% to 25%).

This is a remarkably difficult problem, as it is high-dimensional, highly multimodal, and highly constrained. As shown in Table 7 and Table 8, the best NPV found in 25 runs equals 290.5559; the median NPV equals

263.6527; the mean NPV equals 267.3696; and the worst NPV equals 241.2921; with a σs = 13.1455. The search used up all 10000 time-steps permitted in 23 out of 25 runs. Only 1 run found the best NPV (290.5559); 8 runs found an NPV around 280; 3 runs found an NPV around 270; 10 runs found an NPV of around 260; and the remaining 3 runs found lower NPVs.

Further analysis on the data extracted from the search

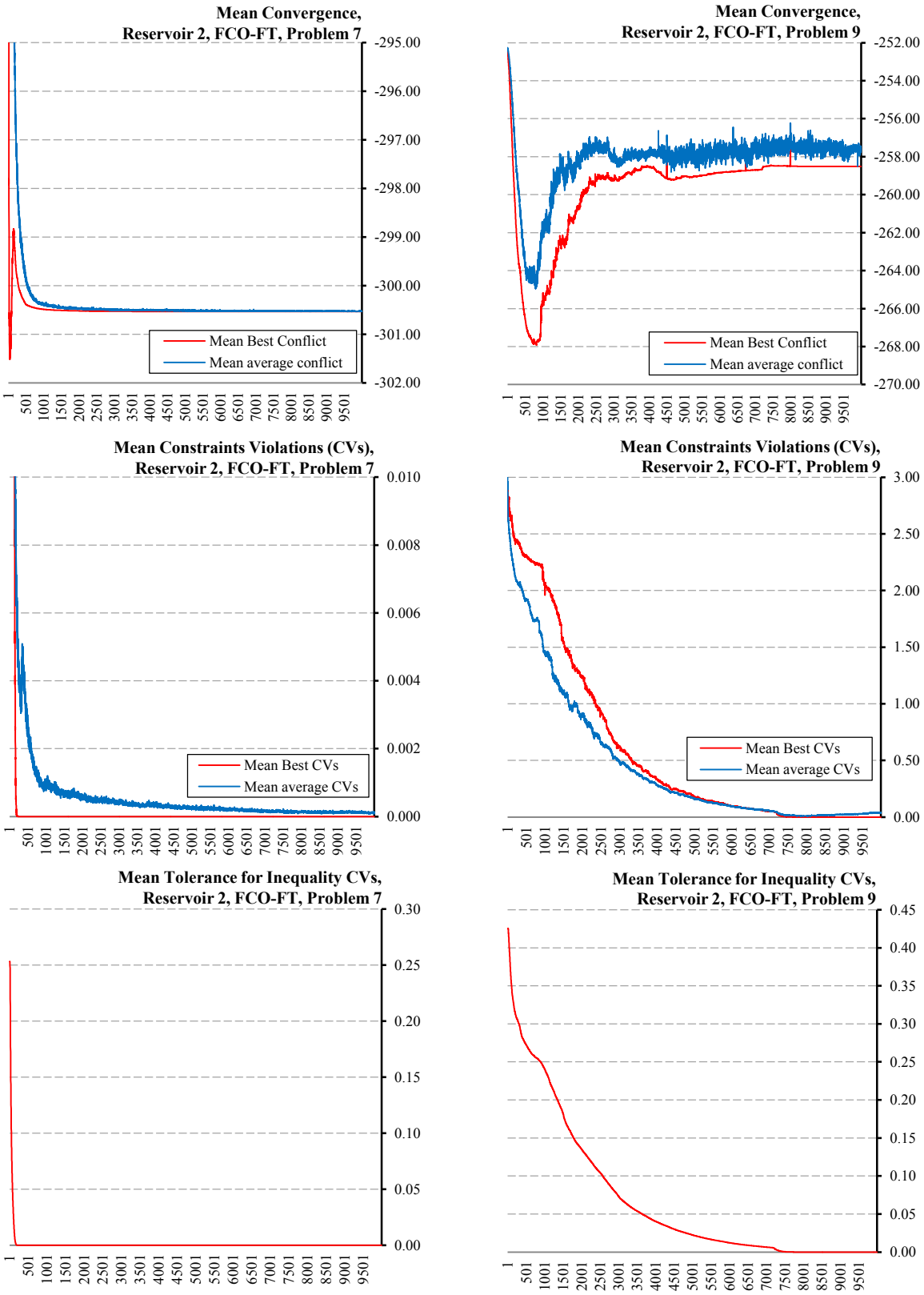

Fig. 11. Mean convergence, mean CVs, and mean tolerance curves for Problems 7 and 9, where conflict stands for $-NPVx10^{-6}$, average is calculated among all feasible particles in a given time-step of a given run, and mean is among 25 runs.

shows that different runs return different NPVs in different regions of the search-space, suggesting that numerous local optima exist in the KM.

As shown in Table 8, the best known solution of this problem (346.3539) differs from the best NPV returned by our PIT (290.5559) in 16.11% of the former.

An accuracy check was performed by evaluating the HF simulator on the coordinates of the best result found by our PIT, resulting in an error of our KM of 9.42% of the HF simulation. Another accuracy check was carried out by evaluating the KM on the coordinates of the best known solution, resulting in an error of our KM of 40.86% of the best known solution (see Table 8).

8. Discussion

For the first reservoir, 4 operational conditions were considered, namely FCO-FT; NFCO-FT; FCO-VT and NFCO-VT (see section 7.1).

Thus, *Problems 1*, *5* and *6* were given by FCO-FT operational condition with 2, 12 and 24 control cycles, respectively. These problems were unconstrained, with 2, 12 and 24 variables, respectively. For each of them, the proposed optimizer found the same Net Present Value (NPV) in every run, and did so quickly. The same performance was shown by all other global optimizers tested on our Kriging models (KMs). Thus, the proposed optimizer performs consistently well on these problems. The KMs of all 3 problems are very accurate, with errors on the coordinates of the best NPVs found below 0.23% of the High Fidelity (HF) simulations, and on the coordinates of the best known solutions below 0.29% of the latter. Overall, the Proposed Integrated Tool (PIT) found best results that differ from the best known solutions in less than 0.15% of the latter.

Problem 2 was given by FCO-VT operational condition with 5 control cycles, 1 inequality constraint, 9 variables and FR = 79.85%. The proposed optimizer converged to the same NPV in every run, and did so reasonably quickly. The same was true for the Multi-Start SQP. The other global optimizers tested on our KMs showed good performance, though not as good. Thus, the proposed optimizer performs consistently well on this problem, with the adaptive tolerance relaxation being beneficial even for high FRs. Furthermore, the KM is accurate, with errors on the coordinates of the best NPV found of 2.02% of the HF simulation, and on the coordinates of the best known solution of 0.76% of the latter. Overall, the PIT found a best result that differs from the best known solution in 1.55% of the latter.

Problem 3 was given by NFCO-FT operational condition with 3 control cycles, 9 inequality constraints, 9 variables and FR = 0.0087%. The proposed optimizer consistently converged to the same region in the search space in 24 out of 25 runs, although the fine-tuning of the search is not as accurate as for the previous problems. Termination conditions other than maximum search-length were never met. The Multi-Start SQP exhibited marginally better performance on our KM converging to the best NPV in every run, whereas the other global optimizers tested showed worse performance. Thus, the proposed optimizer performs consistently well on this problem, though not as well as the Multi-Start SQP, with the adaptive tolerance relaxation being beneficial. The KM is less accurate, with errors on the coordinates of the best NPV found of 7.53% of the HF simulation, and on the coordinates of the best known solution of 6.09% of the latter. Overall, the PIT found a best result that differs from the best known solution in 0.52% of the latter.

Problem 4 was given by NFCO-VT operational condition with 3 control cycles, 10 inequality constraints, 11 variables and FR = 0.008%. The proposed optimizer consistently converged to the same NPV in 23 out of 25 runs, even though termination conditions other than maximum search-length were not met in 24 runs. The Multi-Start SQP exhibited marginally better performance on our KM, converging to the best NPV in every run, whereas the other global optimizers tested showed worse performance. Thus, the proposed optimizer performs consistently well on this problem, though not as well as the Multi-Start SQP, with the adaptive tolerance relaxation being beneficial. The KM is accurate in the vicinity of the best NPV found, with an error of 0.68% of the HF simulation. The KM is less accurate in the vicinity of the best known solution, with an error of 12.97% of the latter. Overall, the PIT found a best result that differs from the best known solution in 0.25% of the latter.

For the second reservoir, 2 operational conditions were considered, namely FCO-FT and NFCO-VT.

Problem 7 was given by FCO-FT operational condition with 1 control cycle, 4 inequality constraints, 10 variables and FR = 0.0794%. The proposed optimizer consistently converged to virtually the same NPV in every run. Termination conditions other than maximum search-length were never met. The Multi-Start SQP exhibited marginally better performance on our KM, converging to the best NPV in every run, whereas the other global optimizers tested showed worse performance. Thus, the proposed optimizer performs consistently well on this problem, though not as well as the Multi-Start SQP, with the adaptive tolerance relaxation being beneficial. The KM is not very accurate in this case, with errors on the coordinates of the best NPV found of 5.09% of the HF simulation, and on the coordinates of the best known solution of 12.59% of the latter. Overall, the PIT found a best result that differs from the best known solution in 1.09% of the latter.

Problem 8 was given by NFCO-VT operational condition with 3 control cycles, 10 inequality constraints, 38 variables and FR = 0.3391%. For this high-dimensional, highly multimodal, and highly constrained problem, the performance of our optimizer deteriorates markedly, returning best NPVs in different regions of the search-space for different runs. The Multi-Start SQP exhibits better performance, although it is also unable to converge to the same NPV in every run on our KM. The other global optimizers tested return worse results. Thus, no optimizer tested consistently converged to the same NPV in every run, with the Multi-Start SQP performing best and our PIT performing second best. The KM is not very accurate, with errors on the coordinates of the best NPV found of 9.42% of the HF simulation, and on the coordinates of the best known solution of 40.86% of the latter. Overall, the PIT found a

best result that differs from the best known solution in 16.11% of the latter.

Finally, *Problem 9* was given by FCO-FT operational condition with 6 control cycles, 24 inequality constraints, 60 variables and FR < 10^{-4}%. For this problem, the Multi-Start SQP converged to the same NPV in every run whereas the other 3 global optimizers returned similarly less accurate NPVs. Among these, our proposed optimizer returns marginally worse results, where 4 out of 25 runs converged to the region of the search-space where the best NPV was found whereas the others converged to different regions displaying conflicts in the range from 252m to 256m. The KM is accurate in the vicinity of the NPV returned by the proposed optimizer, with an error of 1.03% of the HF simulation. The KM is less accurate in the vicinity of the best known solution, with an error of 12.32% of the latter. Overall, the PIT found a best result that differs from the best known solution in 1.67% of the latter.

It is important to highlight that the result returned by the proposed optimizer in every one of the 25 runs for every one of the 9 KMs is better than the corresponding evaluation of the KM on the coordinates of the best known solution, which explains why the optimizer did not converge there. This can be observed in Table 4 and Table 8, where the worst NPVs returned by the proposed optimizer on the KM for every problem is higher (i.e. better) than the evaluation of the corresponding KM on the coordinates of the best known solution.

Note that local optimizer SQP found the best NPV for *Problems 4* and *7*, a good NPV compared to those returned by the other optimizers for *Problem 8*, but low quality NPVs for problems that proved to be relatively easy for the global optimizers like *Problems 1*, *2*, 5 and *6*, and for tougher ones like *Problems 3* and *9*. Thus, the use of global optimizers is advisable at the expense of considerably higher computational effort. Furthermore, results returned by SQP are overly sensitive to the initial guess for multimodal problems, so that its outputs would be less reliable for problems with unknown solutions.

We would like to emphasize here that the aim of this paper is not to compare optimizers' performances. Thus, only 25 runs are carried out for each experiment. This number of runs, together with the maximum number of objective FEs set to 500000 (50 particles, t_{max} = 10000) are in agreement with the evaluation criteria of the *Congress of Evolutionary Computation* (e.g. CEC 2005, CEC 2006, CEC 2008 and CEC 2010). While this is sufficient to show that the proposed optimizer is able to deal with the problems tested with some consistency, a higher number of runs and statistical tests would be required for a rigorous comparison among stochastic algorithms. Nonetheless, for this specific set of 9 problems and the specific results obtained, it seems evident that the optimizer performing the best in terms of the quality of the results is the Multi-Start SQP, with the second best being the proposed optimizer. It also seems evident that the third best is the proposed PSA without the adaptive tolerance relaxation, while the fourth is the GA and the worst results are obtained by the very efficient SQP algorithm.

9. Conclusions and Future Work

In petroleum engineering, waterflooding (WF) is the most widespread method used to improve oil recovery after primary depletion. When searching for optimum management of the process, computationally expensive reservoir simulations are required. In addition, according to the formulation of the operational conditions, the optimization problem may also be highly multimodal and/or highly constrained. Therefore, finding a global search method and a computationally cheaper alternative to reservoir simulations are of great interest. In this paper, we proposed using a Particle Swarm Algorithm (PSA) with general-purpose (default) settings, adaptive Constraint-Handling Technique (CHT) and suitable termination conditions integrated with a Kriging Model (KM) trained offline using a Latin Centroidal Voronoi Tessellation (LCVT) Design of Experiments (DoE) technique and High-Fidelity (HF) reservoir simulations. The Proposed Integrated Tool (PIT) was applied to 9 problems arising from the management of the WF technique in 2 oil reservoirs with different operational conditions and numbers of control cycles.

With regards to the operational conditions, the Non-Full Capacity Operation with Variable Time (NFCO-VT) leads to the highest Net Present Values (NPVs) with fewer design variables and control cycles.

With regards to the KMs trained offline, it can be observed that they are reasonably accurate for problems with Full Capacity Operation (FCO) in the first reservoir. For both reservoirs, accuracy decreases for problems with NFCO. It is interesting to note that, in 4 out of the 5 problems where the accuracy of the KMs is poorer, namely *Problems 4*, *7*, *8* and *9*, the accuracy is noticeably better on the coordinates of the best NPV found by our optimizer than on the coordinates of the best known solutions of the actual problems. For the remaining problem (*Problem 3*), the accuracy of the KM is similar on both locations. This suggests that the optimizer converged towards the better approximations. In other words, the highest accuracies and the highest NPVs of the KMs are located in the same regions.

With regards to the proposed optimizer (PSA + Adaptive CHT), it quickly converged to the same result on every problem with FCO and high Feasibility Ratio (FR) in the first reservoir, irrespective of dimensionality (*Problems 1*, *2*, *5* and *6*). Note that these are also the problems for which the KMs are most accurate. For the 2 problems with NFCO and very low FRs in the first reservoir, the optimizer did not manage to find the exact same result in every run but nearby results of similar quality, thus still showing consistency. For the 10-dimensional problem with FCO and low FR in the second reservoir, the optimizer again converged to the same result in every run. However, the performance deteriorates for the other 2 problems, which are high-dimensional, highly multimodal and highly constrained. Despite the optimizer finding good results, consistency is remarkably poorer than for the other 7 problems.

With regards to the Adaptive Constraint Handling Technique, its incorporation resulted in smaller sample standard deviations (σs) for every problem tested, and led to undoubtedly better results in highly constrained

Problems 3, *4*, *7* and *8*. It is interesting to observe that the adaptive relaxation of the tolerance was beneficial even for *Problem 2*, which presents a very high FR of approximately 80%. For *Problem 9*, the incorporation of the adaptive tolerance relaxation did not lead to clear improvement. In this problem, Fig. 11 shows that the adaptive scheme was unable to drive the tolerance to its desired value smoothly and had to eventually force it. This results in the swarm finding it more difficult to maintain good candidate solutions between tolerance updates. Therefore, the rules for the adaptation of the tolerance relaxation need to be studied further.

With regards to the PIT as a whole, which is composed of a PSA with general-purpose settings, sub-swarms with complementary behaviour, an adaptive CHT and an integrated static KM, it was shown that it is able to obtain near-optimal results without the hassle of performing costly and complex numerical trial-and-error testing and tuning on a case-by-case basis. However, performance deteriorates for increasing dimensionality and for NFCO condition, which increases multimodality. While dimensionality, multimodality and low FRs do not seem to pose overly difficult problems by themselves, the PIT finds it difficult to deal with them when they occur simultaneously.

At a higher level of description, the PIT is composed of a function approximator and a global optimizer. Besides their details, results in this paper suggest that integrating these tools is a good strategy to search for the optimal management of the WF technique. Therefore, different surrogate models, training algorithms, DoE techniques and global optimizers may be considered.

Higher accuracy of the surrogate models would noticeably improve performance of the PIT. This could be achieved by increasing the number of sampling points for the training of the model at the expense of higher computational effort. A next cleverer step would consist of implementing an adaptive scheme so that the training of the surrogate model takes place as the optimization progresses. This would increase the accuracy of the model in the regions of interest only, without spending computational effort in training the model away from the global optimum. This adaptive scheme would also eliminate the need for a DoE technique.

Finally, our aim is to develop a global optimizer for real-world applications that require no parameter tuning or sensitivity analysis. While a first step was to develop an adaptive CHT and termination conditions which control the search-length and hence the computational effort spent according to the difficulty posed by the problem at hand, the use of general-purpose settings for the PSA is a temporary solution. Moving towards a fully adaptive scheme is the next logical step.

Acknowledgements

The second author acknowledges the financial support given by CNPq (National Research Council, Brazil), PETROBRAS, and Foundation CMG.

The other authors acknowledge the support from the Convergence European Regional Development Fund through the Welsh Government via the ASTUTE project.

References

[1] D. W. Zhao, J. Wang and I. D. Gates, "Optimized solvent-aided steam-flooding strategy for recovery of thin heavy oil reservoirs," *Fuel*, vol. 112, pp. 50-59, 2013.

[2] B. Horowitz, S. M. B. Afonso and C. Mendonça, "Surrogate Based Optimal Waterflooding Management," *Journal of Petroleum Science and Engineering* , pp. vl. 112, pp. 206–219, 2013.

[3] C. Chen, G. Li and A. Reynolds, "Robust Constrained Optimization of Short and Long-Term Net Present Value for Closed-Loop Reservoir Management, SPE 141314-PA," in *SPE Journal, Vol. 17, No. 3, pp. 849-864.*, 2012.

[4] D. F. Oliveira and A. Reynolds, "An Adaptive Hierarchical Algorithm for Estimation of Optimal Well Controls, SPE 163645," in *SPE Reservoir Simulation Symposium*, Texas, USA, 18-20 February, 2013.

[5] N. Alexandrov, J. Dennis Jr, R. Lewisand and V. Torczon, "A Trust Region Framework for Managing the Use of Approximation Models in Optimization," *Structural Optimization*, vol. 15, pp. 16-23, 1998.

[6] M. Eldred, A. Giunta and S. Collis, "Second-Order Corrections for Surrogate-Based Optimization with Model Hierarchies, Paper AIAA-2004-4457," in *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, NY, 2004.

[7] M. Powel, "Algorithms for Nonlinear Constraints that use Lagrangian Function," *Mathematical Programming*, vol. 14, no. 1, pp. 224-248, 1978.

[8] L. E. Zerpa, N. V. Queipo, S. Pintos and J.-L. Salager, "An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates," *Journal of Petroleum Science and Engineering*, vol. 47, pp. 197-208, 2005.

[9] A. Forrester, A. Sobester and A. Keane, Engineering Design via Surrogate Modelling: A Practical Guide, Wiley, 2008.

[10] A. Giunta, " Use of Data Sampling, Surrogate Models, and Numerical Optimization in Engineering Design. Paper AIAA-2002-0538," in *40th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, 2002.

[11] IMEX User´s Guide, Computer Modeling Group Ltd, 2012.

[12] S. M. B. Afonso, B. Horowitz and R. B. Willmersdorf, "Comparative Study of Surrogate Models for Engineering Problems," in *7th International ASMO-UK/ISSMO conference on Engineering Design Optimization*, Bath-UK, 2008.

[13] F. A. DiazDelaO and S. Adhikari, "Gaussian process emulators for the stochastic finite element method," *International Journal for Numerical Methods in Engineering*, vol. 87, no. 6, pp. 521-540, 2011.

[14] T. W. Simpson, T. M. Mauery, J. K. Korte and F. Mistree, "Kriging Models for Global Approximations in simulation-Based Multidisciplinary Design Optimization," *AIAA Journal*, pp. 2233-2241, vol 39, 2001.

[15] A. Giunta, S. Wojtkiewicz and M. Eldred, "Overview of Modern Design of Experiments Methods for Computational Simulations. paper AIAA-2003-0649," in *Proceedings of the 41st AIAA Aerospace Sciences Meeting and Exhibit,*, Reno, NV, 2003.

[16] Q. Du, V. Faber and M. Gunzburger, "Centroidal Voronoi Tesselations: Applications and Algorithms," *SIAM Review*, pp. 637–67, vol 41, 1999.

[17] J. Kennedy and R. C. Eberhart, "A discrete binary version

of the particle swarm algorithm," in *Proceedings of the Conference on Systems, Man, and Cybernetics*, Piscataway, 1997.

[18] M. Clerc, "Discrete Particle Swarm Optimization, illustrated by the Traveling Salesman Problem," in *New Optimization Techniques in Engineering (Studies in Fuzziness and Soft Computing)*, Springer-Verlag, 2004, pp. 219-238.

[19] T. Takahama and S. Sakai, "Solving Constrained Optimization Problems by the epsilon Constrained Particle Swarm Optimizer with Adaptive Velocity Limit Control," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS)*, Guangzhou, 2006.

[20] G. Venter and R. T. Haftka, "Constrained particle swarm optimization using a bi-objective formulation," *Structural and Multidisciplinary Optimization*, no. 40, pp. 65-76, 2010.

[21] C. A. Coello Coello, "Constraint-handling through a multiobjective optimization technique," in *Proceedings of the 1999 genetic and evolutionary computation conference, workshop program*, Orlando, 1999.

[22] M. Reyes-Sierra and C. A. Coello Coello, "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287-308, 2006.

[23] C.-k. Chow and H.-T. Tsui, "Autonomous agent response learning by a multi-species particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2004)*, 2004.

[24] S. A. Khan and A. P. Engelbrecht, "A fuzzy particle swarm optimization algorithm for computer communication network topology design," *Applied Intelligence*, vol. 36, no. 1, pp. 161-177, 2012.

[25] K. Khalili-Damghani, A.-R. Abtahi and M. Tavana, "A New Multi-Objective Particle Swarm Optimization method for solving Reliability Redundancy Allocation Problems," *Reliability Engineering & System Safety*, 2012.

[26] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, 1995.

[27] C. Pelley, M. S. Innocente and J. Sienz, "Memetic Particle Swarm for Continuous Unconstrained and Constrained Optimization Problems," in *Proceedings of the 2011 International Conference on Swarm Intelligence (ICSI 11)*, Cergy, 2011.

[28] M. M. Noel, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum," *Applied Soft Computing*, vol. 12, pp. 353-359, 2012.

[29] J. Kennedy and R. C. Eberhart, Swarm Intelligence, Morgan Kaufmann Publishers, 2001.

[30] M. Clerc, Particle Swarm Optimization, Iste, 2006.

[31] R. Poli, J. Kennedy and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, no. 1, pp. 33-57, 2007.

[32] J. Kennedy, "How it works: Collaborative Trial and Error," *International Journal of Computational Intelligence Research*, vol. 4, pp. 71-78, 2008.

[33] M. S. Innocente, Development and testing of a Particle Swarm Optimizer to handle hard unconstrained and constrained problems (Ph.D. Thesis), Swansea: Swansea University, 2010.

[34] J. Sienz and M. S. Innocente, "Individual and Social Behaviour in Particle Swarm Optimizers," in *Developments and Applications in Engineering Computational Technology*, Stirlingshire, Saxe-Coburg Publications, 2010, p. 103–126.

[35] J. Kennedy, "The behavior of particles," in *Evolutionary Programming VII: Proceedings of the 7th Annual Conference on Evolutionary Programming*, San Diego, CA, Berlin: Springer-Verlag, 1998, p. 581–589.

[36] E. Ozcan and C. K. Mohan, "Particle Swarm Optimization: Surfing the Waves," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Washington, DC, 1999.

[37] M. Clerc and J. Kennedy, "The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation, Vol. 6, No. 1,* pp. 58-73, 2002.

[38] I. C. Trelea, "The particle swarm optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters 85,* pp. 317-325, 2003.

[39] F. van den Bergh and A. P. Engelbrecht, "A study of particle swarm optimization particle trajectories," *Information Sciences 176*, no. 176, pp. 937-971, 2006.

[40] M. S. Innocente and J. Sienz, "Coefficients' Settings in Particle Swarm Optimization: Insight and Guidelines," in *Mecom-Cilamce 2010. Mecánica Computacional*, Buenos Aires, Argentina, 2010.

[41] A. Nickabadi, M. M. Ebadzadeh and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing*, vol. 11, pp. 3658-3670, 2011.

[42] M.-S. Leu and M.-F. Yeh, "Grey particle swarm optimization," *Applied Soft Computing*, vol. 12, pp. 2985-2996, 2012.

[43] M.-S. Leu, M.-F. Yeh and S.-C. Wang, "Particle swarm optimization with grey evolutionary analysis," *Applied Soft Computing*, vol. 13, pp. 4047-4062, 2013.

[44] J. Kennedy, "Small Worlds and Mega-Minds: Effect of Neighbourhood Topology on Particle Swarm Performance," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 1999.

[45] R. Mendes, Population Topologies and Their Influence in Particle Swarm Performance (Ph.D. Thesis), Universidade do Minho, 2004.

[46] J. Kennedy and R. Mendes, "Neighbourhood Topologies in Fully-Informed and Best-of-Neighbourhood Particle Swarms," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. Vol.6, no. No.4, pp. 515-519, 2006.

[47] A. Abraham, H. Liu and T.-G. Chang, "Variable Neighborhood Particle Swarm Optimization Algorithm," in *GECCO 2006*, 2006.

[48] A. Mohais, Random Dynamic Neighbourhood Structures in Particle Swarm Optimisation (Ph.D. Thesis), University of the West Indies, 2007.

[49] V. Miranda, H. Keko and A. J. Duque, "Stochastic Star Communication Topology in Evolutionary Particle Swarms (EPSO)," *International Journal of Computational Intelligence Research*, 2008.

[50] J. Kennedy, "Methods of Agreement: Inference Among the EleMentals," in *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, Gaithersburg, 1998.

[51] C. A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. Vol. 41, pp. 113-127, 2000.

[52] P. W. Jansen and R. E. Perez, "Constrained structural design optimization via parallel augmented Lagrangian particle swarm optimization approach," *Computers and Structures*, no. 89, pp. 1352-1366, 2011.

[53] G. Toscano Pulido and C. A. Coello Coello, "A Constraint-Handling Mechanism for Particle Swarm Optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, Portland, 2004.

[54] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, vol. 1, pp. 173-194, 2011.

[55] J. Sienz and M. S. Innocente, "Particle Swarm Optimization: Fundamental Study and its Application to Optimization and to Jetty Scheduling Problems," in *Trends in Engineering Computational Technology*, Stirlingshire, Saxe-Coburg Publications, 2008, pp. 103-126.

[56] F. van den Bergh and A. P. Engelbrecht, "Cooperative Learning in Neural Networks using Particle Swarm Optimizers," *South African Computer Journal*, vol. 26, pp. 84-90, 2000.

[57] C. DeBao and Z. ChunXia, "Particle swarm optimiz ation with adaptive popula tion size and its application," *Applied Soft Computing*, vol. 9, pp. 39-48, 2009.

[58] S. C. Satapathy, S. Chittineni, S. M. Krishna, J. V. R. Murthy and . P. P. Reddy, "Kalman particle swarm optimized polynomials for data classification," *Applied Mathematical Modelling*, no. 36, pp. 115-126, 2012.

[59] A. M. Abdel Tawab, M. B. Abdelhalim and S.-D. Habib, "Efficient multi-feature PSO for fast gray level object-tracking," *Applied Soft Computing*, vol. 14, pp. 317-337, 2014.

[60] K. Mahadevan and P. S. Kannan, "Comprehensive learning particle swarm optimization for reactive power dispatch," *Applied Soft Computing*, vol. 10, pp. 641-652, 2010.

[61] A. Windisch, S. Wappler and J. Wegener, "Applying Particle Swarm Optimization to Software Testing," in *Proceedings of the 2007 Genetic and Evolutionary Computation Conference (GECCO 2007)*, London, 2007.

[62] R. Poli, "Analysis of the Publications on the Aplications of Particle Swarm Optimization," *Journal of Artificial Evolution and Applications*, 2008.

[63] A. P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons Ltd, 2005.

[64] M. R. AlRashidi and M. E. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Electric Power Systems," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 913-918, 2009.

[65] D. Swagatam, A. Ajith and K. Amit, "Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives," in *Advances of Computational Intelligence in Industrial Systems*, vol. 116, L. Ying, S. Aixin, L. HanTong, L. WenFeng and L. Ee-Peng, Eds., Springer, 2008, pp. 1-38.

[66] V. P. Plagianakos, D. K. Tasoulis and M. N. Vrahatis, "A Review of Major Application Areas of Differential Evolution," in *Advances in Differential Evolution*, vol. 143, U. K. Chakraborty, Ed., 2008, pp. 197-238.

[67] R. M. Calazan, N. Nedjah and L. M. Mourelle, "A hardware accelerator for Particle Swarm Optimization," *Applied Soft Computing*, vol. 14, pp. 347-356, 2014.

[68] R. Hassan, B. Cohanim, O. De Weck and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, 2005.

[69] E. Elbeltagi, T. Hegazy and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, pp. 43-53, 2005.

[70] D. Bratton and J. Kennedy, "Defining a Standard for Particle Swarm Optimization," in *Proceedings of the 2007 IEEE Swarm Intelligence Symposium (SIS 2007)*, 2007.

[71] J. J. Liang and P. N. Suganthan, "Dynamic Multi-Swarm Particle Swarm Optimizer," in *Proceedings of the IEEE International Swarm Intelligence Symposium (SIS 2005)*, 2005.

[72] T. Blackwell and J. Branke, "Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 459-472, 2006.

[73] R. Poli, "Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers During Stagnation," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 4, pp. 712-721, 2009.

[74] J. C. Fuentes Cabrera and C. A. Coello Coello, "Handling Constraints in Particle Swarm Optimization Using a Small Population Size," in *MICAI 2007: Advances in Artificial Intelligence*, 2007, pp. 41-51.

[75] T. Takahama, S. Sakai and N. Iwane, "Solving Nonlinear Constrained Optimization Problems by the epsilon Constrained Differential Evolution," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, Taipei, 2006.

[76] M. S. Innocente and J. Sienz, "Pseudo-Adaptive Penalization to Handle Constraints in Particle Swarm Optimizers," in *Proceedings of the 10th International Conference on Computational Structures Technology*, Valencia, 2010.

[77] X. Hu, R. C. Eberhart and Y. Shi, "Engineering Optimization with Particle Swarm," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, Indianapolis, 2003.

[78] D. F. B. Oliveira, Otimização da Produção para Reservatórios de Petróleo: Abordagens Sem Uso de Derivadas para Técnicas de Alocação Dinâmica das Vazões de Produção e Injeção, Master Thesis (in Portuguese), Recife, Brazil: Civil Engineering Department, UFPE, 2006.

[79] L. C. Oliveira, Hibrid Strategy Applied to the optimum reservoir management (Ph.D. Thesis), Recife: Federal University of Pernambuco, 2013.

[80] F. Silva, J. Rodrigues, P. Paraizo, R. Romeu, A. Peres, R. Oliveira, I. Pinto and C. Maschio, "Novel Ways of Parameterizing the History Matching Problem, SPE 94875," in *SPE Latin and Caribbean Petroleum Engineering Conference* , Rio de Janeiro, Brazil, 2005.