Petroleum 2 (2016) 79–89

Contents lists available at ScienceDirect

Petroleum

journal homepage: www.keaipublishing.com/en/journals/petlm

Original article

On the application of artificial bee colony (ABC) algorithm for optimization of well placements in fractured reservoirs; efficiency comparison with the particle swarm optimization (PSO) methodology

Behzad Nozohour-leilabady [a], Babak Fazelabdolabadi [b,*]

[a] *Department of Petroleum Engineering, International Campus of the University of Tehran, Kish Island, Iran*
[b] *Institute of Enhanced Oil Recovery, Centre of Exploration and Production Studies and Research, Research Institute of Petroleum Industry (RIPI), Tehran, Iran*

ARTICLE INFO

Article history:
Received 1 September 2015
Received in revised form
1 November 2015
Accepted 3 November 2015

Keywords:
Artificial bee colony (ABC)
Particle swarm optimization (PSO)
Well placement

ABSTRACT

The application of a recent optimization technique, the artificial bee colony (ABC), was investigated in the context of finding the optimal well locations. The ABC performance was compared with the corresponding results from the particle swarm optimization (PSO) algorithm, under essentially similar conditions. Treatment of out-of-boundary solution vectors was accomplished via the Periodic boundary condition (PBC), which presumably accelerates convergence towards the global optimum. Stochastic searches were initiated from several random staring points, to minimize *starting-point dependency* in the established results. The optimizations were aimed at maximizing the Net Present Value (NPV) objective function over the considered oilfield production durations. To deal with the issue of reservoir heterogeneity, random permeability was applied via normal/uniform distribution functions. In addition, the issue of increased number of optimization parameters was address, by considering scenarios with *multiple* injector and producer wells, and cases with *deviated wells* in a real reservoir model. The typical results prove ABC to excel PSO (in the cases studied) after relatively short optimization cycles, indicating the great premise of ABC methodology to be used for well-optimization purposes.

1. Introduction

Development of robust methodologies for computational optimization of oil-field operations has been the matter of extensive research in recent years. In this regards, maximization of profits through optimizing future production is considered as a crucial topic. The issue is generally treated in a dual context; namely, finding the optimal well settings/controls, and determining the best well locations. The two aspects are often treated in a separate manner, albeit their intricate relationship [1]. In the latter context, the problem involves determination of type, number, and/or location of new wells (to be drilled).

Such a problem is computationally demanding in the sense that the impact of reservoir heterogeneity (e.g. permeability field) has to be incorporated, in addition to analyzing a large number of development scenarios. The practical approach for this issue entails numerous (objective) function evaluations, each requiring a full set of reservoir simulation run. Particularly in well-placement problems, the objective function considered is that of the Net Present Value (NPV). The NPV objective function generally holds a *multi-modal* nature, as being *non-convex* and *non-smooth* with several local optima [2], and exhibits a rougher surface compared to its well-control objective function counterpart [2,3]. This adds extra difficulty to the well-placement case, even though the reservoir simulations in this category are often performed under the (straightforward) conditions of fixed flow-rate/bottom-hole pressure (BHP) at injection/production wells. A stochastic framework is often devised to tackle the issue

* Corresponding author. Tel.: +98 21 48252148.
E-mail address: fazelb@ripi.ir (B. Fazelabdolabadi).
Peer review under responsibility of Southwest Petroleum University.

Production and Hosting by Elsevier on behalf of KeAi

http://dx.doi.org/10.1016/j.petlm.2015.11.004

in the well-placement case. Typically, stochastic techniques are more efficient in globally searching the solution space, and can avoid converging prematurely to local optima, as opposed to deterministic methods. The excellence of the stochastic technique used will therefore play a pivotal role in the overall success of this computational challenge.

Different stochastic techniques have been applied to the well-placement problem, including simulated annealing [2], Genetic algorithm (GA) [4–7], simultaneous perturbation stochastic approximation (SPSA) [8], particle swarm optimization (PSO) [9,10], and covariance matrix adaptation evolution strategy (CMA-ES) [5,11]. Combination of stochastic techniques with deterministic approaches/proxy models gave rise to the formation of hybrid-stochastic algorithms, which were also applied to the well-placement optimization [12,13]. The main competition in this regards has been on finding the best global optimum in its shortest computational time, or equivalently least reservoir simulation runs.

Artificial bee colony (ABC) is a relatively new stochastic algorithm for global optimization. The algorithm mimics the intelligent foraging behavior of honey bee swarm. It is categorized into the swarm-based class of the population-based optimization algorithms. ABC is capable of handling unconstrained [14,15], as well as constrained optimization problems [16]. Since its recent inception, the ABC algorithm has been used in a vast pool of applications, such as neural network training [17,18], cluster analysis [19], protein structure prediction [20], and stock market forecasting [21]. Recent numerical studies have unveiled the ABC's distinguished capability, to excel other comparable meta-heuristic algorithms, in continuous optimization problems. Karaboga and Akay [22], for instance, used the ABC technique for optimizing a large set of complicated benchmark test functions, and compared the results with the corresponding results obtained from other well-known population-based techniques, such as Evolution Strategy (ES), Differential Evolution (DE), GA, and PSO. In all numerical cases studied, the ABC had either exhibited better performance or produced similar results, with an added benefit of employing fewer control parameters [22]. The ABC technique was also found to out-perform the meta-heuristic Firefly algorithm (FA) [23] in optimization tasks, but was surpassed in some test problems by the Cuckoo-search (CK) algorithm [24]. Moreover, Hybrid-ABC algorithms have been proposed by combining the ABC concept with the GA/DE strategies [25–27], producing efficient optimization results. In spite of its widespread usage in computational community, the ABC methodology has rarely been applied to oilfield development, and its mere usage so far has been limited to training the neural network framework for BHP prediction in underbalanced drilling [28]. This fact served as our principal motivation in using the ABC concept for discovering optimal well locations.

We will give an in-depth description of our ABC methodology in the next section. In Section 3; we will present our ABC results and compare the performance with the corresponding PSO results. This will be ensued by our conclusions.

2. ABC algorithm

The ABC algorithm is tailored to simulate the foraging behavior of a honeybee colony. A typical honeybee swarm consists of three fundamental components: food source/employed foragers/unemployed foragers (bees). Employed foragers are the bees that are employed at, and currently exploiting, a certain food source. They carry information about the (distance, direction and the profitability) of the food source and communicate the information with other bees waiting at the hive. Unemployed bees are classified as being either an onlooker bee, or as a scout bee. The former tries to find a food source by means of the information given by an employed bee; while the latter randomly searches the environment to find a new (better) food source [22]. Presumably, an employed bee whose food source is depleted becomes a scout bee, and starts to search for a new food source. Furthermore, it assumes the number of employed bees in the colony to be equal to the number of food sources. Conceivably, the position of a food source represents a possible solution to the optimization problem; whereas the amount of a food source corresponds to the quality (fitness) of the associated solution.

Initially, the ABC generates a randomly distributed population of *SN* solutions (food source positions) in the search space, where *SN* denotes the size of employed bees or onlooker bees [22]. Assuming the number of optimization parameters to be *D*, then each solution $x_i (i = 1, 2, \ldots, SN)$ will essentially be a *D*-dimensional vector. All solutions generated at this stage can be obtained from Eq. (1) [22]:

$$x_{ij} = x_{\min,j} + rand[0,1]\left(x_{\max,j} - x_{\min,j}\right) \tag{1}$$

Here, x_{min} and x_{max} are respectively the lower and upper-boundary parameters for solution x_i in dimension j ($j = 1, 2 \ldots D$), and $rand[0,1]$ is a scaling factor representing a random number between [0,1]. The *D*-dimensional solutions (food source positions) generated in the initialization step ($C = 0$) are subject to repeated cycles ($C = 1, 2 \ldots, MCN$), until a termination criterion is satisfied. Both global as well as local probabilistic search/selection are implemented in a single ABC cycle. Each cycle entails a number of tasks performed by different bee types. These operations are essentially independent, and can be explained in a separate manner as follows, for better elucidation of the ABC methodology:

2.1. Employed bee tasks

After being assigned to their food sources, the employed bees evaluate the fitness of their sources (solutions) and communicate the information with the onlooker bees. In addition, each employed bee generates a candidate solution (food position) by perturbing the old solution (x_{ij}) in its memory, using the expression below [22]:

$$v_{ij} = x_{ij} + rand[-1,+1]\left(x_{ij} - x_{kj}\right) \tag{2}$$

Here, $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, SN\}$ ($k \neq i$) are randomly-chosen indexes, and $rand[-1,+1]$ is a random number between [−1,1], which works as a scaling factor. It is evident that as the optimum solution is approached in the search space, this perturbation on solution gets reduced. The fitness of the new (perturbed) solution will also be evaluated by employed bee, and in case better fitness values are found, the new solution replaces the old one in the memory of that employed bee (a *greedy-selection* scheme).

2.2. Onlooker bee tasks

The primary task of an onlooker bee is to select a food source (solution), based on the probability value associated with that food source, P_i, which is evaluated by the following expression [26]:

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

where *fit* represents the fitness value of a given solution, and the subscript index refers to the solution number. This probabilistic selection is implemented by comparing *Pi* against a randomly chosen number between [0,1]. The selection is approved if the generated random number is less than or equal to *Pi*, and will be rejected otherwise. Assignment of an onlooker bee to a given solution will be sanctioned, if the corresponding probabilistic selection is approved. Typically, in minimization problems, the fitness value of solutions is calculated by the following expression:

$$fit_i = \begin{cases} \frac{1}{1+f_i} & (f_i \geq 0) \\ 1+|f_i| & (f_i < 0) \end{cases} \tag{4}$$

where f_i is the value of the objective function for solution *i*. Having selected a food source with a P_i probability, an onlooker bee will choose a new food source (solution) in the neighborhood of the previous one in her memory, using Eq. (2). Should the new solution have a better fitness value, an onlooker bee will then updates the new solution in her mind, and forgets the old one, similar to the case with employed bees.

2.3. Scout bee tasks

The task of scout bees is to randomly explore the *entire* search space, so as to find a new (improved) solution to the global optimization problem. Unlike the case with employed/onlooker bees (which were essentially bounded to generate trial solution *around* an old solution), the scout bees are unbounded in that sense. They draw their samples from an extensive set of *D*-dimensional vectors, as long as it remains inside the search space boundaries. In ABC, if a (non-global) solution cannot be improved further after a pre-determined number of cycles, then that solution will be abandoned, and the employed bee assigned to that particular position will convert to a scout bee with essentially scout-type functionality. The value of this pre-determined number of cycles, which is termed the *limit*, will therefore be an important control parameter in the algorithm. In practice, the limit is estimated via the following expression:

$$limit = c \cdot n_e \cdot D \tag{5}$$

where n_e is the number of unemployed bees, and *c* is a constant coefficient with a recommended value of 0.5 or 1 [14,15]. At least one scout bee should be present during ABC implementation. In fact, scout-type operations provide outstanding capability to the ABC method in finding the best global solution, by making stochastic searches in the entire *D*-dimensional space. In other words, scout bees will independently search for a global optimum solution, while other bee types (employed/onlooker) are simultaneously examining their local candidate solutions for the global best. In that sense, the possibility of being trapped in local optima will never be applicable to ABC.

2.4. Implementation of ABC to well placement optimization

The ABC framework can be applied to the well-placement optimization problem, by devising the following algorithm [22]:

Algorithm 2.1 ABC algorithm

1: Generate a randomly-distributed population of solutions (x_i) in the search space. $i \in \{1,2,...,SN\}$

2: Evaluate the NPV objective function of solutions in step 1.

3: cycle = 1

4: **Repeat**

5: Produce new solutions (v_i) for employed bees by using Eq.2 and evaluate their NPV values.

6: For employed bees, apply the greedy selection scheme (between v_i and x_i).

7: Calculate the probability value P_i for the solution x_i .

8: Assign onlooker bees to the solutions with probabilistic selection.

9: Produce new solutions (v_i) for onlooker bees by using Eq.2 and evaluate their NPV values.

10: Apply the greedy selection process to onlooker bees.

11: Determine the abandoned solution for the scout, if exists, and replace it with a randomly-produced solution (x_i) by using Eq.1.

12: Produce new solutions (x_i) for scout bees, if no solution was abandoned.

13: Memorize the best solution, so far.

14: cycle = cycle + 1

15: **Until** cycle=Maximum number of cycles

The solution vectors in this algorithm pertain to different specifications for well locations. Also the greedy-selection scheme here is based on selecting the solution vector with correspondingly higher NPV value. Roughly 5–10% of the bee population is devoted to scout-type bees [22].

2.5. Treatment of out-of-boundary vectors

As mentioned earlier, consideration of solution vectors for NPV calculations is only valid if the vector resides inside the search-space domain. However, at some instances during the ABC algorithm implementation (steps 5, 9, 11, 12) the possibility exists that some of the trial solution vectors exceed the search-space boundaries. A rudimentary treatment of this issue is to simply disregard the out-of-bound vectors, or to place them back *at* the boundaries. This approach is inefficient, as we will either lose some searching vectors, or will end up having most of our searching vectors being positioned at the boundaries of the domain. Alternative way is provided by molecular simulation concepts [29]. It entails considering a Reflective Boundary Condition (RBC), or a Periodic Boundary Condition (PBC) for handling the out-of-boundary vectors. Specific to our optimization problem, the RBC/PBC implementation can be described as follows. For RBC, if a vector exceeds the system boundaries, then it will be placed back *inside* the domain at a position which is equally-distanced to the boundary as the exceeding point, but in an opposite direction. In PBC, if a vector exceeds the system boundaries, it will be placed back *inside* the domain at a position which is equally-distanced to the boundary as the exceeding point, but the entrance is made in the same direction, but from the other corresponding end of the system. Fig. 1 provides an illustration of the two techniques.

Mathematical expression for PBC moves are as follows:

$$x_{ij} = \begin{cases} x_{\max,j} - \left|x_{ij} - x_{\min,j}\right| & (x_{ij} < x_{\min,j}) \\ x_{\min,j} + \left|x_{\max,j} - x_{ij}\right| & (x_{ij} > x_{\max,j}) \end{cases} \tag{6}$$

2.6. Evaluation of objective function

Optimization processes generally aim at finding the global optimum of an objective function. The objective function considered in the bulk of well-placement studies (including the present work) is that of the NPV. Estimation of NPV value of each candidate solution vector (i.e. well location) requires the fluid production data, which is obtained by running a distinct reservoir simulation:

$$NPV = \sum_{t=1}^{T} \frac{CF_t}{(1+r)^t} - C^{capex} \tag{7}$$

Here, T is the total production time in years, r is the annual discount rate, C^{capex} is the capital expenditure, and CF_t is the cash flow at time t. C^{capex} is defined as the total cost of drilling and completing all of the wells at time $t = 0$, evaluated as follows [9]:

$$C^{capex} = \sum_{\omega=1}^{N^{well}} \left[C_{\omega}^{top} + L_{\omega}^{main} C^{drill} + \sum_{l=1}^{N_{\omega}^{lat}} \left[C_l^{junc} + L_{l,\omega}^{lat} C^{drill} \right] \right] \tag{8}$$

where, N^{well} is the number of wells, N_{ω}^{lat} is the number of laterals in well ω, C_{ω}^{top} is the cost of drilling the main bore to the top of the reservoir ($), C^{drill} is the per-foot cost of drilling within the reservoir ($/ft), C_l^{junc} is the junction cost of lateral l ($), L_{ω}^{main} is the length of the main bore (ft), and $L_{l,\omega}^{lat}$ is the length of lateral l (ft). CF_t is estimated by the value of revenue/operating expenditure as:

$$CF_t = R_t - E_t \tag{9}$$

where R_t is the revenue ($), and E_t is the operating cost ($) of the project. Fluid production data is incorporated into estimation of both quantities, as follows:

$$R_t = p_O Q_t^O + p_g Q_t^g \tag{10}$$

$$E_t = p_w^p Q_t^{w,p} + p_w^i Q_t^{w,i} \tag{11}$$

Here, p_O is the oil price ($/STB), p_g is the gas price ($/SCF), Q_t^O is the total volume of produced oil at time t (STB), Q_t^g is the total volume of produced gas at time t (SCF), p_w^p is the water production cost ($/STB), p_w^i is the water injection cost ($/STB), $Q_t^{w,p}$ is the total volume of water produced at time t (STB), and $Q_t^{w,i}$ is the total volume of injected water at time t (STB). These expressions (Eqs. (7)–(11)) will hold generality and can be applied to problems with different well types and to the case with varying drilling costs.

The economic parameters used for NPV estimation in the present work is listed in Table 1. Only vertical wells were considered ($C_l^{junc} = 0$, $L_{l,\omega}^{lat} = 0$) with no water injection ($p_w^i = 0$) scenario.

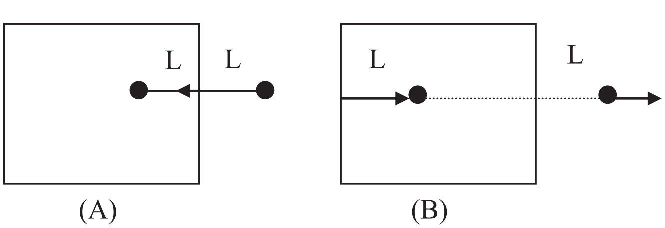

Fig. 1. Treatment of out-of-boundary point in RBC (A) and PBC (B).

3. Results of optimization runs

In this section, we present our ABC results for well-placement optimization. Three set of examples were considered. The first two sets involved finding optimized well locations for cases with one and ten wells, respectively. The third example related to the case of optimizing the well location for three deviated producer wells (using the real field data) in a fractured reservoir. For all cases studied, we obtained the corresponding PSO optimization results, for later comparison with the ABC. The treatment of out-of-boundary solution vectors was tested under both RBC/PBC. Yet the PBC provided faster convergence towards finding the global optimum, and the reported results here have been developed under the PBC. Our ABC/PSO optimization codes were written in the R programming language [30], which has a proven record of excellence for statistical computing.

Table 1
Economic parameters for NPV estimation.

Drilling cost to reservoir top, C_{ω}^{top}	50×10^6 ($)
Drilling cost peer foot, C^{drill}	10,000 ($/ft)
Oil price, p_O	100 ($/STB)
Gas price, p_g	5 ($/SCF)
Water production cost, p_w^p	5 ($/STB)
Annual discount rate, r	0.1

3.1. Example-1: a single-well location optimization case

With an aim to evaluate the performance of ABC method against reservoir heterogeneity, we first tested the ABC performance in finding the optimized well locations for a single producing well case, in an (artificial) reservoir case. The reservoir considered consisted of 30 × 30 × 5 grid blocks, with each block being 200 × 200 × 50 ft. in size. Two scenarios were devised, as for the reservoir heterogeneity. In the first scenario (S-1), random permeability was assigned to each grid block via a normal distribution function. Whereas in the second scenario (S-2), the assignment of random permeability values to each grid block was accomplished through a uniform distribution function. The selection of random permeability values in the S-1 and S-2 cases was made by using the *rnorm* and *runif* functions in the R programming language, respectively. For the S-1 case, the values were derived from a normal distribution function with a mean ($\mu = 400$) and standard deviation ($\sigma = 12.66$); this ensures the 0.001 and 0.998 quantiles of the distribution function to be at 360.85 and 436.45, respectively. The choice of our parameters for mean and standard deviation was arbitrary. For the S-2 case, the values were extracted from a uniform distribution function in the arbitrary range of [0.01, 500] millidarcy. The number of random numbers derived in each case, was obviously equal to the number of grid blocks considered. Our procedure entailed selecting a list of randomly-distributed numbers, with a list size equal to the number of grid blocks in the system. Each number in the list was then assigned to a grid block, in the sequel (i.e. the first number to the first grid block, etc.).

Within each scenario, two (different) realizations were generated and tested with ABC/PSO. Fig. 2 depicts the random permeability fields considered for each realization, in Example-1.

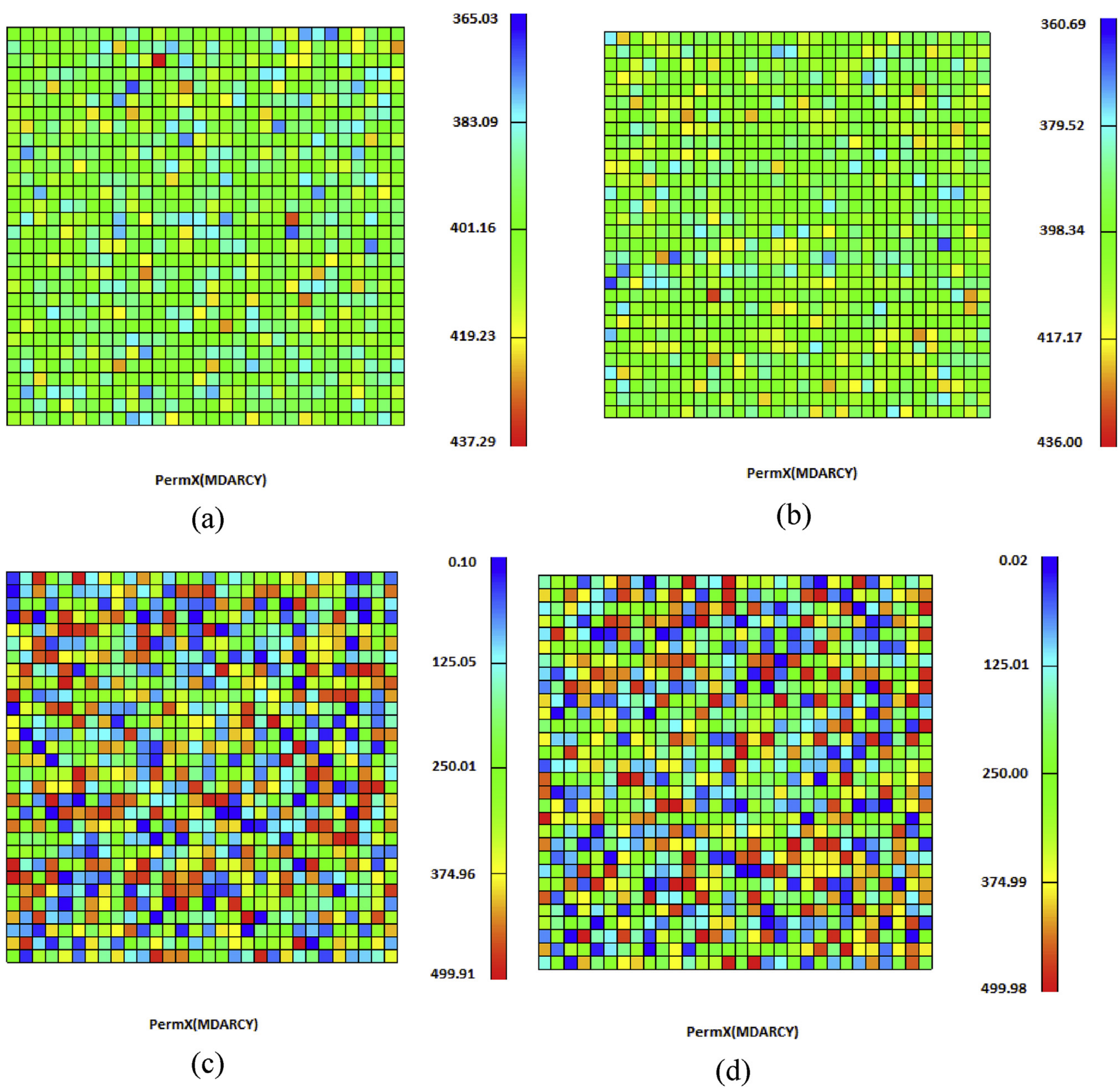

Fig. 2. Random permeability in Example-1, for (a) & (b) cases with permeability values obtained from a normal distribution function, and (c) & (d) cases with permeability values selected from a uniform distribution function.

The porosity in all the grid blocks was taken constant, to be equal to 0.2, and the production time considered was for 1250 days. The general parameters used in reservoir simulations are listed in Table 2.

For each realization, (assuming the well to be perforated in only a single layer) there will be 30 × 30 possibilities for positioning a single well in this field. Therefore, for each realization in Example-1, we can construct the entire objective function surface by performing 900 reservoir simulation runs. This was accomplished and the result is presented in Fig. 3; nevertheless, for the sake of brevity, only the results pertaining to Fig. 2a/c are reported herein. In Example-1, the variables (to be optimally selected) were the x–y coordinates, or alternatively the *ii* and *jj* indices of the grid block, in which the single vertical well should be drilled (see Fig. 3). The objective function to be maximized in Example-1 was then the NPV (Eq. (7)).

As evident from the figures, the permeability difference between adjacent grid blocks in S-2 case is more conspicuous than the corresponding S-1 situation; subsequently causing sharp changes amongst the resultant NPV values for S-2 case (as shown in Fig. 3b). Such a situation should expose great challenge to an optimization technique to overcome.

Both ABC and PSO algorithms were used for optimization (under essentially similar conditions). The PSO was implemented, according to the guidelines proposed elsewhere [11]. The optimization runs were attempted from several random staring points, to minimize the *starting-point dependency* of results.

For our S-1/S-2 cases in Example-1, the result of our 900 sets of simulations, point to global NPV values of $371204457488 and $362970565157, respectively.

The issue of susceptibility of results to the swarm size (i.e. number of searching particles), or the number of optimization cycles, was also investigated in the present analysis. Fig. 4 shows the established results, for the ABC/PSO optimization runs. To enhance accuracy, each specific case was run for 30 times, and the results are presented (Fig. 4) on the average of the thirty results obtained at a given swarm size/maximum optimization cycles for ABC/PSO.

As evident from Fig. 4, the performance of ABC has excelled PSO, after typically 3–4 optimization cycles in each scenario. In addition, ABC has been able to detect the global optimum in relatively short optimization cycles, or yielded a better final NPV value. In S-2 case, with essentially higher irregularity in NPV surface, the ABC has exhibited a comparably successful record with small swarm size.

3.2. Example-2: ten vertical wells location optimization cases

In this example, we considered a slightly more complicated case, to find the optimal well locations for ten vertical wells (including eight injector and two producer wells) in a reservoir with 50 × 50 × 3 grid blocks. Dealing with increased number of wells, will inevitably test the efficiency of an optimization technique to deal with an added number of optimization parameters.

Each grid block, in Example-2, was 100 × 100 × 50 ft. in size. The reservoir properties and the simulation control parameters were essentially similar to Example-1. The total production time considered was 1440 days. Moreover, it was assumed that each well was perforated at only a single layer.

Analogous to Example-1, the variables (to be optimized) were the x–y coordinates of the optimal placements of the ten vertical wells; or alternatively the *ii* and *jj* indices of the grid blocks, in which the ten vertical well should are to be drilled (see Fig. 3). The objective function to be maximized in Example-2 was also that of the Net Present Value (Eq. (7)).

Random permeability field was applied in this Example-2; however, to add more complexity to the case, the permeabilities were selected from a uniform distribution function. Similar to Example-1, each optimization run was repeatedly tested, each

Table 2
General parameters used in reservoir simulations in Examples 1 & 2.

Phase	Oil, gas, water
Depth of reservoir top	5000 (ft)
Well radius	0.5 (ft)
Oil density	49.94 (lb/ft^3)
Water density	62.43 (lb/ft^3)
Gas density	0.061 (lb/ft^3)
Pressure at the depth of 5040 ft	3600 (psi)
Water formation volume factor[a]	1 (STB/bbl)
Water viscosity[a]	0.5 (cP)
Rock compressibility[a]	4×10^{-6} (psi^{-1})

[a] Properties at the reference pressure of 4000 (psi).

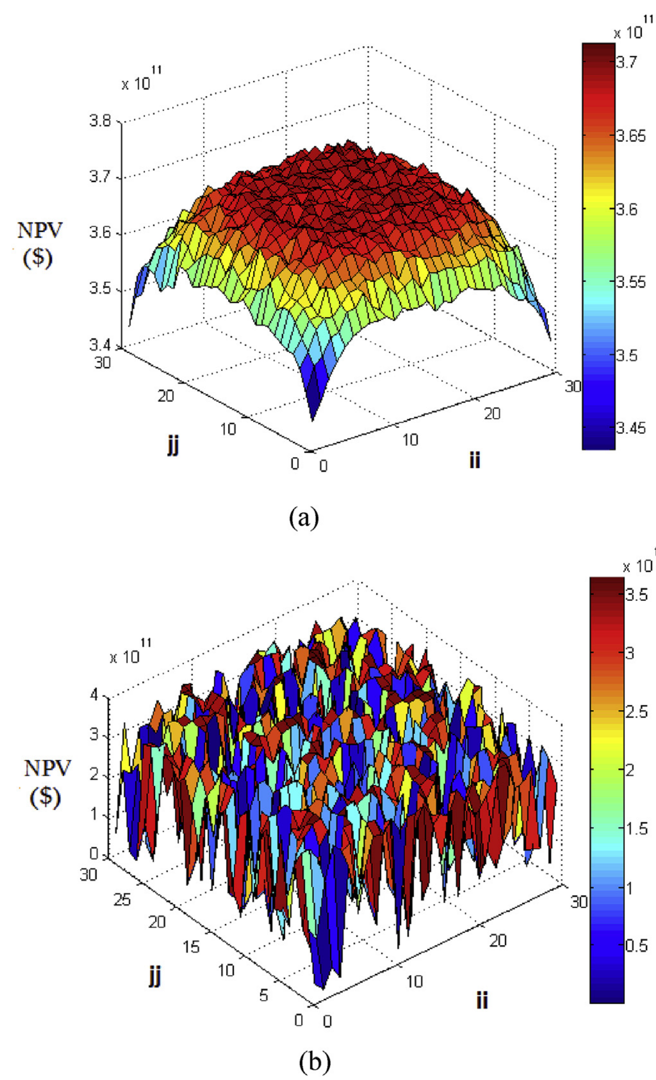

Fig. 3. NPV surface for Example-1, for (a) case with permeability values obtained from a normal distribution function (Fig. 2a), and (b) case with permeability values selected from a uniform distribution function (Fig. 2c).

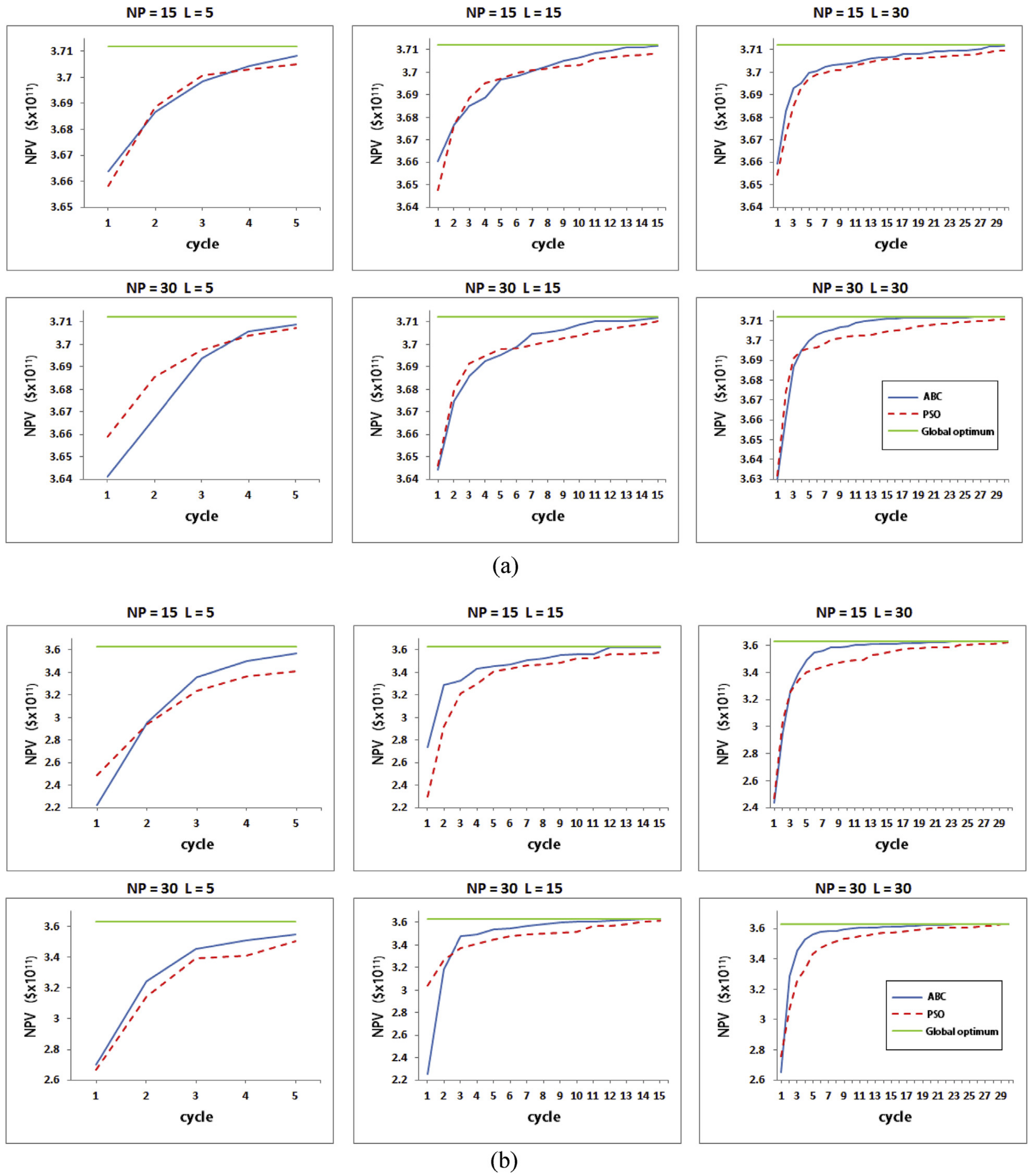

Fig. 4. NPV versus the number of optimization cycle for (a) S-1 (b) S-2 cases of Example-1, at different swarm sizes, *NP*, and different maximum number of optimization cycles, *L*.

time starting from a random starting point, and the average results were reported. Table 3 enlists the parameters used during the optimization runs, in this example.

For improved accuracy, a number of five different optimization runs were performed for each ABC/PSO case, and the average results are presented in Fig. 5.

The average NPV value obtained for ABC/PSO cases were $150659975433 and $139571340075, respectively. The optimum

Table 3
Parameters used during ABC/PSO optimization runs.

Example	Max. cycles	ABC		PSO			
		# Employed bees	Limit	c_1[a]	c_2[a]	w[a]	# Particles
Example-2	100	30	30	1.108	1.108	0.721	30

[a] Velocity-term coefficients, according to reference [9].

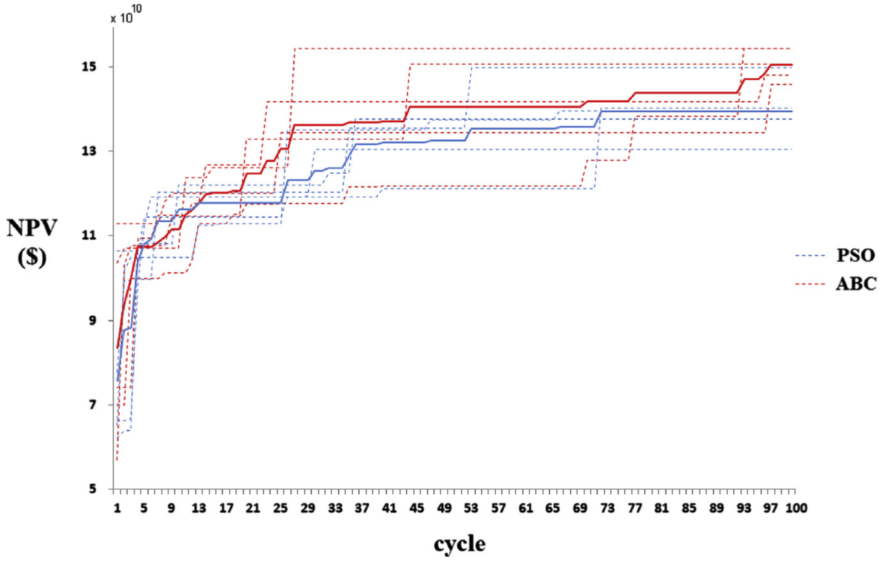

Fig. 5. NPV of the best solutions as a function of the number of optimization runs for PSO (blue curves) and ABC (red curves) for Example-2. *Thin lines* correspond to individual runs, while *thick lines* represent averages over the five runs.

well locations (for eight producer and two injector wells) obtained is shown in Fig. 6. Similar to Example-1, in this example, the results indicate the (average) performance of ABC has surpassed the corresponding PSO after relatively short optimization cycles, and sustained excellence towards the end of optimization runs (Fig. 5).

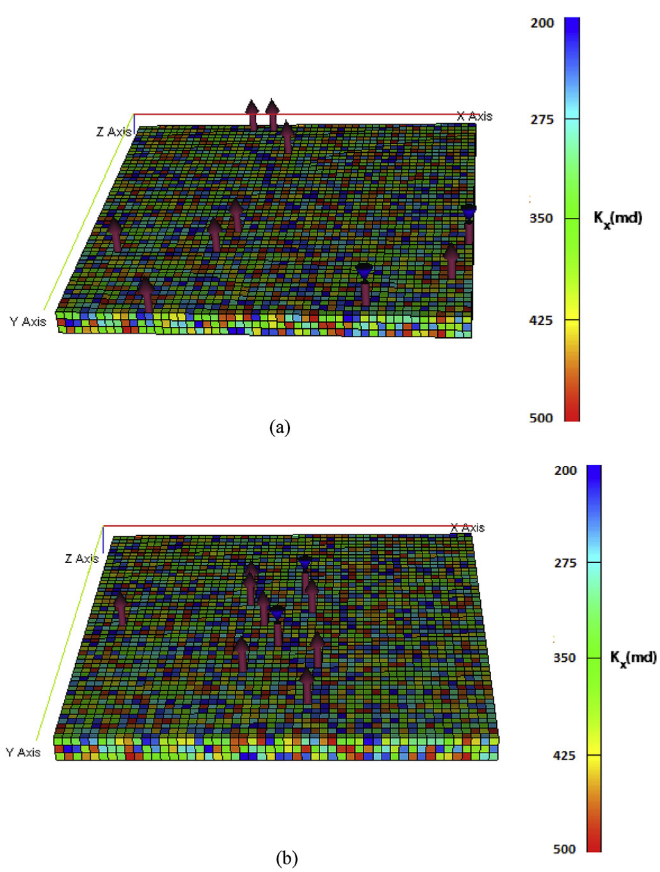

Fig. 6. The optimum well locations (of eight producer and two injector wells) in Example-2, for (a) ABC and (b) PSO cases.

3.3. *Example-3: the optimum well placement of three deviated producer wells in a real fractured reservoir*

In order to test the ABC's performance in problems with higher degree of complexity, we applied the technique to a model based on real data from a fractured reservoir. The issue in this example related to an essentially more complex situation of

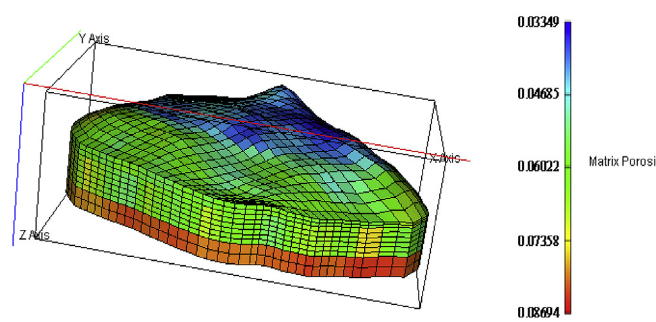

Fig. 7. Porosity distribution of the reservoir in Example-3.

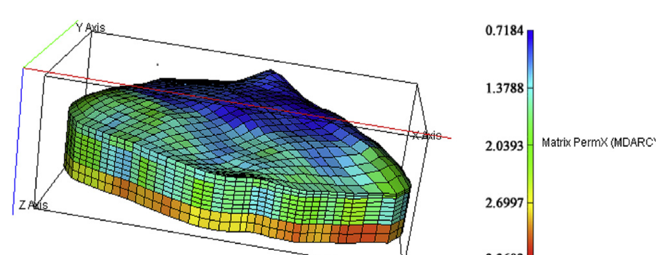

Fig. 8. Permeability distribution of the reservoir in Example-3.

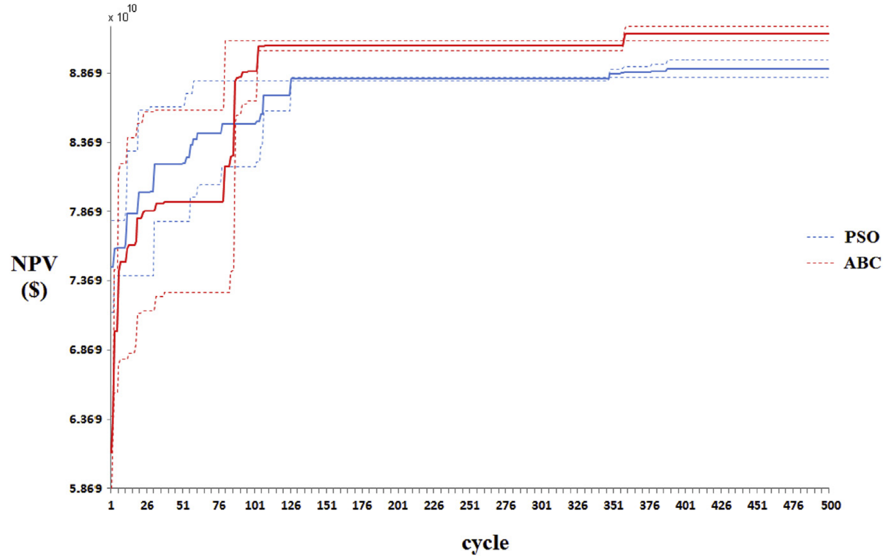

Fig. 9. NPV of the best solutions as a function of the number of optimization runs for PSO (blue curves) and ABC (red curves) for Example-3. *Thin lines* correspond to individual runs, while *thick lines* represent averages over the two runs.

dealing with a *deviated* well, as opposed to vertical wells in the previous two examples.

The model considered in Example-3, consisted of $30 \times 30 \times 10$ grid blocks, with the actual porosity/permeability distributions shown in Figs. 7 and 8, respectively. The time duration of production from this reservoir was 2160 days.

The reservoir pressure (at a depth of 900 ft.) is 4100 psi. At this pressure, the water viscosity was 0.5 (cp). Moreover, the compressibility of rock/water was taken as 1.244×10^{-6} and 2.1×10^{-6} psi^{-1}, respectively. In addition, the deviated drilling cost per foot within the reservoir was considered at 1,00,000 ($/ft). Additional geological information about the reservoir considered in Example-3 is provided in the Supplementary Information to the present article. The datum depth, water-oil-contact (WOC), and gas-oil-contact (GOC) in this example, were taken as 900 (ft), 1950 (ft), and 600 (ft), respectively.

A set of *six* optimization parameters were devised for each *deviated* well, comprising a total of 18 optimization parameters for the three deviated wells considered. The six variables considered under each scenario in Example-3 were those of (x^{top}, x^{junc}, x^{bottom}, y^{top}, y^{junc}, y^{bottom}). Here x and y refer to the x and y coordinates of the grid block, respectively. The superscripts *top*, *junc*, and *bottom* refer to the top, junction, and bottom points of the deviated well being considered. Likewise the previous two examples, the NPV is considered as our objective function to be optimally maximized. The whole optimization runs were repeated *twice*, and the results shown in Fig. 8 pertain to the average of the corresponding two results for ABC/PSO (Fig. 9) (Table 4).

Table 4
Parameters used during ABC/PSO optimization runs, in Example-3.

Example	Max. cycles	ABC		PSO			
		# Employed bees	Limit	$c_1^{\ddagger}$	$c_2^{\ddagger}$	$w^{\ddagger}$	# Particles
Example-3	500	8	24	1.108	1.108	0.721	8

In Example-3, the averaged NPV for the PSO and ABC optimization runs were $89024231918 and $91558021616, respectively. This clearly indicates the ABC methodology to surpass PSO, for the sake of finding the optimal well locations of the three deviated wells studied in Example-3 (Fig. 10). In a similar trend to the previous two examples, the NPV results also convey the fact that ABC's performance (on average) will excel PSO after relatively few iterations, and the difference can be appreciable as the situation involves more complexity.

4. Concluding remarks

In the present work, we applied the novel ABC optimization algorithm to find the optimal locations of vertical/deviated wells in different models of random permeability. In general, ABC showed outstanding record in cases with reservoir heterogeneity (in which the random permeability was generated from normal/uniform distribution functions). The excellence of ABC sustained in situations with added degree of complexity, in terms of optimization parameters. Such *increased-parameter* cases were devised by *either* increasing the number wells, *or* by considering deviated wells. The ABC methodology was equally superior in finding optimal locations in scenarios with both injector and producer wells. The scaling behavior of ABC results was nearly unaltered with swarm size, in the cases analyzed. In general, both ABC/PSO exhibit good performances for well placement purposes. Nevertheless, in the examples studied, the ABC was found to excel the PSO in finding the global optimum, either in fewer computational cycles *on average*, or (in some cases) was the only optimum detecting technique in the computational range considered. Such excellence of ABC was verified to be *starting-point independent*. In the course of our optimization processes, the PBC type of treatment of out-of-boundary solution vectors, was found to yield better contribution towards convergence to the optimal. We conclude, the ABC methodology should hold great promises to be applied to field-scale optimizations.

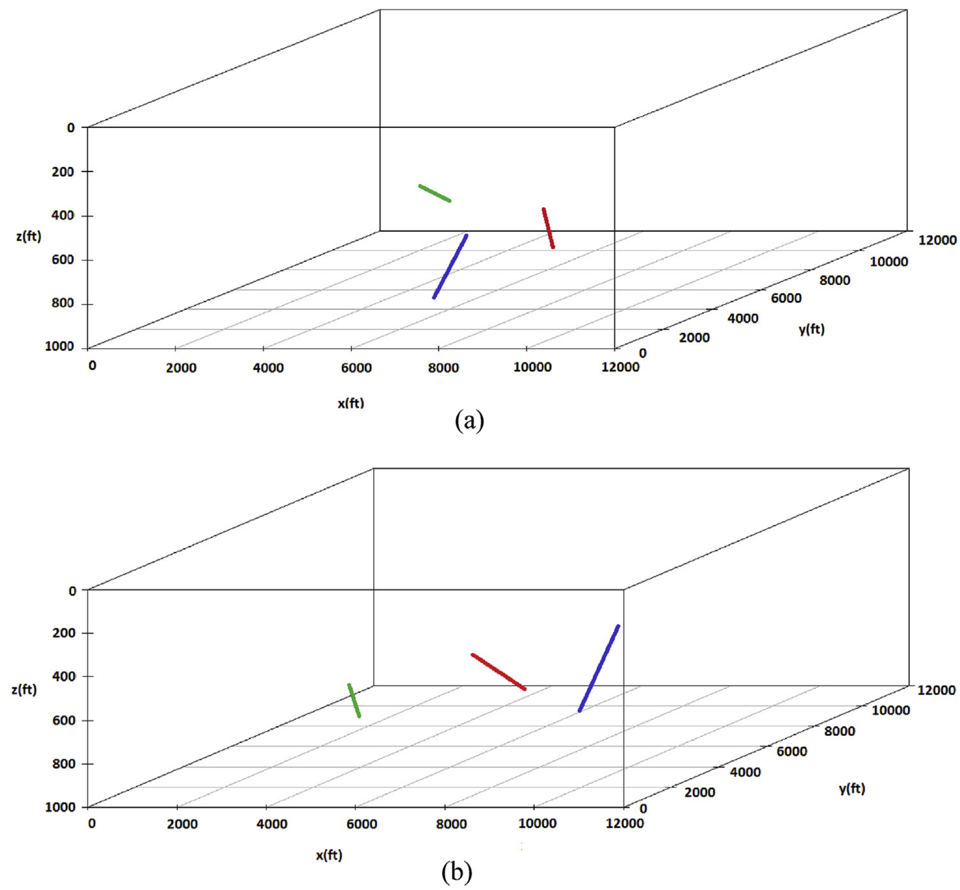

Fig. 10. Three-dimensional view of the optimal well locations detected by (a) ABC, (b) PSO methods in Example-3.

Appendix A. Supplementary data

Supplementary data related to this article can be found at http://dx.doi.org/10.1016/j.petlm.2015.11.004.

References

[1] J.E. Onwunalu, L.J. Durlofsky, Application of a particle swarm optimization algorithm for determining optimum well location and type, Comput. Geosci. (2010), http://dx.doi.org/10.1007/s10596-009-9142-1.
[2] B. Guyagular, R.N. Horne, Uncertainty assessment of well placement optimization, SPE Reserv. Eval. Eng. (2004), http://dx.doi.org/10.2118/87663-PA.
[3] W. Bangerth, H. Klie, M.F. Wheeler, P.L. Stoffa, M.K. Sen, On optimization algorithms for the reservoir oil well placement problem, Comput. Geosci. (2006), http://dx.doi.org/10.1007/s10596-006-9025-7.
[4] Z. Bouzarkouna, D.Y. Ding, A. Auger, Partially separated meta-models with evolution strategies for well placement optimization, in: SPE EUROPEC/EAGE Annual Conference and Exhibition, 2011, http://dx.doi.org/10.2118/143292-MS.
[5] A. Aliyev, Use of Hybrid Approaches and Metaoptimization for Well Placement Problems (Master's Thesis), Stanford University, 2011.
[6] D. Karaboga, B. Basturk Akay, C. Ozturk, Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks, LNCS Model. Decis. Artif. Intell. (2007), http://dx.doi.org/10.1007/978-3-540-73729-2_30.
[7] B.L. Beckner, X. Song, Field development planning using simulated annealing—optimal economic well scheduling and placement, in: SPE Annual Technical Conference and Exhibition, 1995, http://dx.doi.org/10.2118/30650-MS.
[8] D. Frenkel, B. Smit, Understanding Molecular Simulation from Algorithms to Applications, Academic Press, 2002.
[9] A. Abraham, R.K. Jatoth, A. Rajasekhar, Hybrid differential artificial bee colony algorithm, J. Comput. Theor. Nanosci. (2012), http://dx.doi.org/10.1166/jctn.2012.2019.
[10] E. Nwankwor, A.K. Nagar, D.C. Reid, Hybrid differential evolution and particle swarm optimization for optimal well placement, Comput. Geosci. (2013), http://dx.doi.org/10.1007/s10596-012-9328-9.
[11] B. Yeten, L.J. Durlofsky, K. Aziz, Optimization of nonconventional well type, location, and trajectory, SPE J. (2003), http://dx.doi.org/10.2118/86880-PA.
[12] C.M.V. Benítez, H.S. Lopes, Parallel artificial bee colony algorithm approaches for protein structure prediction using the 3DHP-SC model, Intell. Distrib. Comput. IV (2010), http://dx.doi.org/10.1007/978-3-642-15211-5_27.
[13] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Appl. Math. Comput. (2009), http://dx.doi.org/10.1016/j.amc.2009.03.090.
[14] V. Artus, L.J. Durlofsky, J. Onwunalu, K. Aziz, Optimization of nonconventional wells under uncertainty using statistical proxies, Comput. Geosci. (2006), http://dx.doi.org/10.1007/s10596-006-9031-9.
[15] X. Yan, Y. Zhu, W. Zou, A hybrid artificial bee colony algorithm for numerical function optimization, in: 11th International Conference on Hybrid Intelligent Systems, Melacca, 2011, http://dx.doi.org/10.1109/HIS.2011.6122092.
[16] T.D. Humphries, R.D. Haynes, L.A. James, Simultaneous and sequential approaches to joint optimization of well placement and control, Comput. Geosci. (2013), http://dx.doi.org/10.1007/s10596-013-9375-x.
[17] R. Irani, R. Nasimi, Application of artificial bee colony-based neural network in bottom hole pressure prediction in underbalanced drilling, J. Petrol. Sci. Eng. (2006), http://dx.doi.org/10.1016/j.petrol.2011.05.006.
[18] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. (2007), http://dx.doi.org/10.1007/s10898-007-9149-x.
[19] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. (2008), http://dx.doi.org/10.1016/j.asoc.2007.05.007.
[20] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, LNCS Adv. Soft Comput. Found. Fuzzy Log. Soft Comput. (2007), http://dx.doi.org/10.1007/978-3-540-72950-1_77.
[21] D. Karaboga, B. Akay, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, Appl. Soft Comput. (2011), http://dx.doi.org/10.1016/j.asoc.2010.12.001.
[22] J.E. Onwunalu, L.J. Durlofsky, A new well-pattern-optimization procedure for large-scale field development, SPE J. (2011), http://dx.doi.org/10.2118/124364-PA.
[23] S.R. Khaze, I. Maleki, S. Hojjatkhah, A. Bagherinia, Evaluation the efficiency of artificial bee colony and the firefly algorithm in solving the continuous

optimization problem, Int. J. Comput. Sci. Appl. (2013), http://dx.doi.org/10.5121/ijcsa.2013.3403.

[24] T. Hsieh, H. Hsiao, W. Yeh, Forecasting stock markets using wavelet transforms and recurrent neural networks: an integrated system based on artificial bee colony algorithm, Appl. Soft Comput. (2011), http://dx.doi.org/10.1016/j.asoc.2010.09.007.

[25] D.E. Ciaurri, T. Mukerji, L.J. Durlofsky, Derivative-free optimization for oil field operations, Comput. Optim. Appl. Eng. Ind. (2011), http://dx.doi.org/10.1007/978-3-642-20986-4_2.

[26] P. Civicioglu, E. Besdok, A conceptual comparison of the cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, Artif. Intell. Rev. (2011), http://dx.doi.org/10.1007/s10462-011-9276-0.

[27] Y. Zhang, L. Wu, S. Wang, Y. Huo, Chaotic artificial bee colony used for cluster analysis, Intell. Comput. Inf. Sci. (2011), http://dx.doi.org/10.1007/978-3-642-18129-0_33.

[28] R Core Team, R: a Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2013. URL, http://www.R-project.org.

[29] Z. Bouzarkouna, D.Y. Ding, A. Auger, Well placement optimization with the covariance matrix adaptation evolution strategy and meta-models, Comput. Geosci. (2012), http://dx.doi.org/10.1007/s10596-011-9254-2.

[30] D. Karaboga, B. Basturk Akay, Artificial bee colony algorithm on training artificial neural networks, Signal Process. Commun. Appl. (2007), http://dx.doi.org/10.1109/SIU.2007.4298679.