

Assisted History Matching for Petroleum Reservoirs in the Social Computing Era

¹Michel Cancelliere, ²Dario Viberti and ²Francesca Verga

¹Department of Projects and Technology, Shell Calgary Research Center (CRC), Calgary, Canada

²Department of Environment, Land and Infrastructure Engineering (DIATI) of Politecnico di Torino, Torino, Italy

Received 2013-06-06, Revised 2013-07-16; Accepted 2013-07-22

ABSTRACT

The exploitation strategy of hydrocarbon reservoirs can be technically and economically optimized only if a reliable numerical model of the reservoir under investigation is available to predict the system response for different production scenarios. A numerical model can be reasonably trustworthy after calibration only, which means the model has at least proved its ability to reproduce the historical behavior of the reservoir it represents. The calibration procedure, also known as history matching, is the most time consuming phase in a reservoir study workflow. Over the last decades several methods, classified as Assisted History Matching (AHM), have been proposed for a partial automation of the model calibration procedure. Meta-heuristic methods have been used to iteratively reduce the misfit between simulated and historical data. However, the main limit for the application of these algorithms is the amount of computational time necessary for the evaluation of the objective function, thus for the simulation runs. On the other hand, the new trend on collective computing offers a solution to CPU intensive tasks by distributing the work among several computers located in different places but globally connected through the World Wide Web. In this study a novel workflow for assisted history matching is proposed. The results proved that this workflow provides better and more representative solutions in a fraction of the time needed by traditional approaches.

Keywords: Reservoir Simulation, History Matching, Multi-objective Optimization, Evolutionary Strategies, Collective Computing

1. INTRODUCTION

Petroleum reservoirs contain naturally occurring hydrocarbon systems, which are mixtures of organic compounds exhibiting multiphase behavior over wide ranges of pressures and temperatures. Hydrocarbon accumulations may occur in the gaseous state, in the liquid state or in various combinations of both. Reservoir engineering has the task to study the behavior and the characteristics of an oil or gas reservoir so as to determine the future development and production strategies that could maximize the profit. The forecasts of production scenarios are strongly conditioned by the uncertainty in the parameters that define the reservoir model. The calibration of the model, also known as

History Matching (HM), aims at reducing such uncertainty by simulating the reservoir from its initial state to the current configuration and comparing results with the historical data. Adjustments to the model parameters are made until a satisfactory match is achieved between the simulation results and the past production data. Typically, history matching is carried out with a trial and error approach which is not straightforward and, usually, very time consuming. In fact, the team performing the calibration has to deal with the non-uniqueness issue, as history matching is an ill-posed inverse problem due to insufficient constraints and data (Schaaf *et al.*, 2008). Thus, history matching is by far the most challenging phase of reservoir simulation.

Corresponding Author: Michel Cancelliere, Department of Projects and Technology, Shell Calgary Research Center (CRC), Calgary, Canada

Some years ago, a new methodology called automatic history matching was approached by the scientific community. The idea consisted in treating history matching as an optimization process, i.e., defining an objective function representative of the discrepancy between measured (real) and simulated data and in minimizing the objective function. The minimization of the objective function could be obtained by applying a suitable optimization algorithm. Optimization and non-linear programming were not new methodologies in the field of applied mathematics and more knowledge has been gained more recently. However, the selection of the most adequate optimization algorithm among those available in the technical literature is not trivial and the number of independent variables involved in complex reservoir simulation does not make the solution of the optimization problem a standard procedure. In fact, the main criticalities of a history matching process change for each analyzed reservoir. As a consequence, the identification of an optimization methodology appropriate for a wide variety of reservoirs is rather impossible (Wolpert and Macready, 1997). Therefore, automatic history matching remains a dream of sorts and, more realistically, assisted history matching can be the target. Traditionally, history matching was carried out by reservoir engineers because of their knowledge of the fluid flow and physical phenomena occurring in the porous media. The concept of assisted history matching implies that reservoir engineers are still in charge of reservoir model calibration, but they can rely on optimization tools to better explore the parameter space and to speed up the convergence to one or more solutions.

In the recent past, studies have proven that the use of assisted methods together with the optimization theory can considerably reduce the time needed to calibrate a model (Cullik *et al.*, 2006; Selberg *et al.*, 2006; Fokker *et al.*, 2013). Furthermore, assisted methods can provide multiple possible solutions, which offer a much more representative evaluation of the uncertainty associated with the production forecasts.

In Assisted History Matching (AHM) the simulated dynamic data is compared to the historical data by means of a misfit function. The algorithms try to minimize the misfit function and thus to obtain the model that better approximates the past production data. This procedure can be translated into an optimization problem, bounded by the model constraints, in which the misfit function is an objective function.

Recently, extensive research has been done on ensemble-based methods (e.g., Ensemble Kalman Filter (EnKF), Ensemble Smoother (ES), Randomized Maximum Likelihood (RML) (Oliver and Chen, 2010). Also, meta-heuristics methods based on particle swarm have been proposed (Mohamed *et al.*, 2011). Many of them can be adapted to work with the computational environment proposed in this study. In this work a multi-objective optimization procedure based on the SPEA2 algorithm was adopted.

The objective or misfit function can be expressed as a single-objective function or as a multi-objective function. Traditionally, the single-objective functions have been the most used ones, but they require that the user specifies the weights associated to each set of data to be calibrated. The advantage of the multi-objective functions is that they can simultaneously minimize different kinds of data using the Pareto criterion. Several methodologies and techniques have been studied for such optimization problems (Gomez *et al.*, 2001; Schaaf *et al.*, 2008; Riegert *et al.*, 2001). They can be roughly divided into local and global algorithms. The advantage of local algorithms, such as gradient-based algorithms, is that they converge faster to a minimum than global algorithms. However, they only provide a single solution which is the nearest local optimum to the initial guess (Landa *et al.*, 2005). Conversely, global algorithms can provide multiple solutions in a single run and can escape from local minima efficiently. Heuristic methods such as simulated annealing, Genetic Algorithms (GA) and Evolutionary Strategies (ES) are known to be highly effective searching techniques (Riegert *et al.*, 2001). Nonetheless, they require a large number of evaluations of the misfit function; in most cases these evaluations are represented by simulation runs. Thus, for numerical models with a large number of grid cells the computational cost of heuristic methods can be prohibitive.

Combinations of the above-mentioned methods have also been developed. In particular, the idea of global-local optimization has proven to be very effective because it provides a good trade-off between the exploration of the solution space and the exploitation of possible minima. Sensitivity-based methods, such as Gauss-Newton and the Least Square method (LSQR) have also been tested. The gradient-based method is intrinsically sequential and cannot exploit efficiently parallel architectures; on the other hand, the global optimization methods, in general, are easily parallelizable and can greatly benefit from

distributed architectures, which allow running several simulations simultaneously (Selberg *et al.*, 2006). Current developments in multi-core processors allow parallelization of numerical codes and, as a consequence, speed up of the calculations (Oliveira *et al.*, 2013).

Over the last years, the scientific community has taken a great leap forward in the automation of the history matching process to calibrate dynamic reservoir models. Some of the methods have been inherited from other scientific disciplines; others have been constructed ad-hoc for the history matching problem. But for the moment, there is no clear winner.

In this study, a new workflow based on multi-objective evolutionary strategy was adopted in connection to an in-house black oil simulator based on the finite volume method (Cancelliere and Verga, 2012) for assessing the potential of AHM using the new techniques on collective computing. This workflow simultaneously boosts the integration among disciplines and significantly reduces the time needed for the calibration of a model. Furthermore, it introduces the possibility for the user to improve the optimization procedure by actively interacting to steer the behavior of the algorithms.

2. MATERIALS AND METHODS

Evolutionary Strategies (ES) are meta-heuristic optimization algorithms and, as their name suggests, they are based on the theory of the evolution and behavior of living species.

The theory of evolution is based on the changes that a population of individuals experience from one generation to the next one, supposed to be better or fitter than the previous one. This is due to the action of the main elements of evolution, which are mutation, recombination and selection. The genome of an individual, for example, can be randomly altered by mutation. If the new traits and adaptations are beneficial for the survival of the individual, they are passed on to the next generations. Alterations and new traits in the species are also caused by the mixture of the genetic codes of the parents when generating offspring: this is called recombination. Eventually, natural selection plays a very important role in evolution. In fact, the fittest individuals are more likely to reproduce, thus the offspring will likely be better and fitter individuals than the previous generation. The same concepts have been utilized in optimization problems so as to progressively improve the solution population.

When the evolutionary strategies are applied, the population of candidate solutions for an optimization problem is forced to interact based on the evolutionary criteria described above. The general scope is to obtain several optimal solutions, which represent the fittest individuals, after a given number of iterations. The fitness function is used to evaluate how good the candidate solution is with respect to the optimum. In an Evolutionary Strategy algorithm each individual is represented by a set of real numbers that can be easily manipulated by the evolutionary operators.

In single objective optimization problems the objective function, or cost function, representing the discrepancy between observed and simulated data, is minimized in order to obtain the optimal solution.

In AHM the objective function is usually represented by Equation 1:

$$f(x) = \sum_i a_i \frac{(y_i^*(x) - y_i)^2}{\sigma_i^2} \quad (1)$$

where, for each i index: y_i represents the measurement, $y_i^*(x)$ is the value simulated using the set of decision variables x ; σ_i^2 is the variance and a_i is the assigned weight. The formulation of a single objective optimization process is expressed by Equation 2:

$$\begin{cases} \min f(x) \\ h_k^l \leq x_k \leq h_k^u \\ x = \{x_1, x_2, \dots, x_k, \dots, x_N\} \end{cases} \quad (2)$$

where, $x = \{x_1, x_2, \dots, x_N\}$ is the vector of the N decision variables, i.e., the unknowns of the problem and the variables h_k^l and h_k^u represent the lower and upper boundaries of each unknown.

The solution can be obtained by applying different minimization techniques depending on the characteristics of the objective function. In real hydrocarbon reservoirs, characterized by a large number of wells, strong heterogeneities, uncertainties on historical data, complex geometries, the behavior of the parameters observed at each well, such as gas and water rates, well pressure, can be strongly uncorrelated, making it difficult to define the right set of weighting parameters a_i . Additionally, in several cases different choices of the weighting parameters can lead to different matches that are equally acceptable.

In multi-objective optimization this problem is avoided by splitting the objective function into several functions which are optimized simultaneously and the objective function takes a vector form (Equation 3):

$$F(x) = \left\{ \begin{array}{l} f_1(x) = \frac{(y_1^*(x) - y_1)^2}{\sigma_1^2} \\ \dots, f_M(x) = \frac{(y_M^*(x) - y_M)^2}{\sigma_M^2} \end{array} \right\} \quad (3)$$

The multi-objective optimization problem for an AHM is stated as follows Equation 4:

$$\left\{ \begin{array}{l} \min F(x) \\ h_k^l \leq x_k \leq h_k^u \\ x = \{x_1, x_2, \dots, x_k, \dots, x_N\} \end{array} \right. \quad (4)$$

where, $F(x): \mathbb{R}^N \rightarrow \mathbb{R}^M$ and x, h_k^l and h_k^u have been previously defined.

Since different objectives are not comparable, the concept of optimal solution is substituted by the Pareto-optimal set also called Pareto front. A vector of decision variables $x^* \in \mathfrak{S}$ is Pareto optimal if there is not another $x^* \in \mathfrak{S}$ such that $f_i(x) \leq f_i(x^*)$ for all $i = 1, \dots, k$ and $f_j(x) < f_j(x^*)$ for at least one j (**Fig. 1**). The vectors x^* corresponding to the solutions included in the Pareto-optimal set are called non-dominated.

The modified Strength Pareto Evolutionary Algorithm (SPEA2) is one of the most important multi-objective evolutionary algorithms which use the elitism approach. It was initially described by (Zitzler *et al.*, 2001) as an improved version of the SPEA algorithm (Zitzler, 1999).

In the SPEA2 a raw fitness for each individual is calculated on the basis of the fitness value of the solutions which dominate. The differentiation between individuals having identical raw fitness is accomplished by adding density information to the fitness function. The mating pool in SPEA2 is filled by choosing the best individuals from the archive and from the current population in terms of non-dominance and diversity.

In order to preserve diversity, the SPEA2 applies a truncation procedure that depends on the minimum distance among non-dominated individuals.

In single-objective optimization problems the efficiency of the algorithms can be tested by monitoring the reduction of the objective function with the algorithm iterations. However, in multi-objective optimization processes this test is not feasible due to the presence of several objective functions. A solution is to merge different objectives by using the hyper-volume concept. In the volume-based method the goal is to minimize the fraction of the space (or hyper-volume) which is not dominated by any of the archive members. The non-dominated space is then normalized by using a reference volume between the origin and a reference point, also called utopia point. The advantage of this approach consists in considering the distance of the solutions from the utopia point and the level of clusterization of the solutions simultaneously.

A synthetic case is presented and discussed in the following to prove the effectiveness and flexibility of the multi-objective optimization for AHM, enriched by the possibility to steer the process with the integration of a socially-assisted approach (Verga *et al.*, 2013).

Historical production rates and bottom hole pressures to be matched were generated for a synthetic reservoir model. The simulator was an in-house black oil simulator based on the finite volume method. The use of an in-house simulator allowed bypassing the restrictions on the number of available licenses, which is a relevant limitation (or cost) when working on distributed computation. Having removed this constraint, the number of simulations that can be run simultaneously only depends on the number of available computers.

The reservoir is a heterogeneous and anisotropic oil-bearing formation divided into two layers of different porosity and into eight permeability regions.

The physical domain consisted in a rectangular reservoir of 900×900 m in the horizontal plain and a thickness of 60 m. A Cartesian grid was used with 15 cells in the x and y directions and 2 cells in the vertical direction. No-flow boundaries were assumed at the limits of the physical domain and no aquifer was considered. The porosity was assumed equal to 0.26 for the top layer and 0.18 for the bottom layer.

The absolute permeability values for the different regions are summarized in **Fig. 2**. The different permeability regions are shown in **Fig. 3**. The wells were completed through all the reservoir thickness. Zero mechanical skin was assumed at the wells. The initial pressure of the reservoir at the datum depth of 3000 m was assumed to be 300 bar.

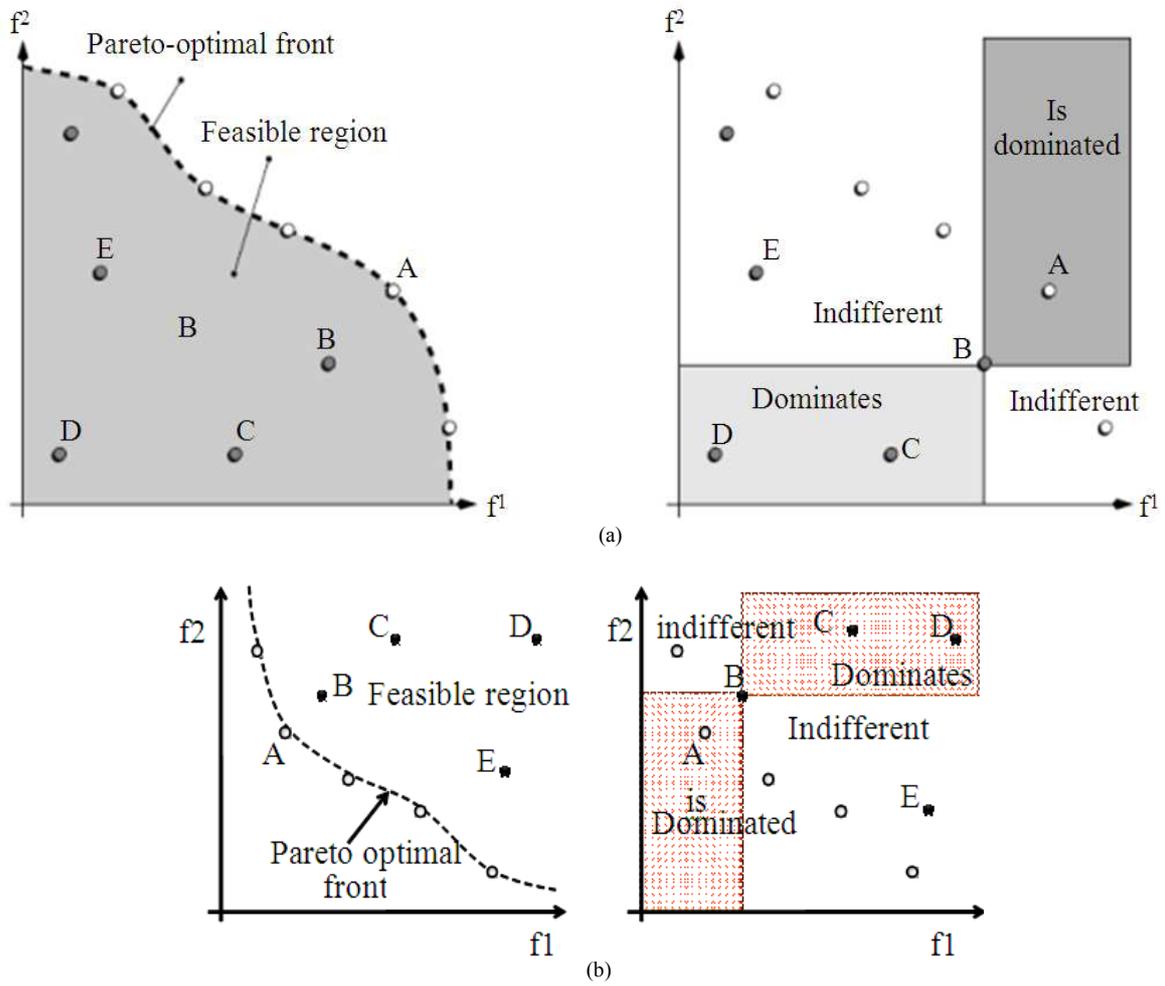


Fig. 1. Pareto-optimal set and Pareto front (for a maximization problem) (Zitzler, 1999)

Top layer		Bottom layer	
PERM1 20mD	PERM2 120mD	PERM1 20mD	PERM6 140mD
PERM3 50mD		PERM 40mD	
PERM4 200 mD	PERM5 25mD	PERM8 80mD	PERM5 25mD

Fig. 2. Permeability maps in the two reservoir layers

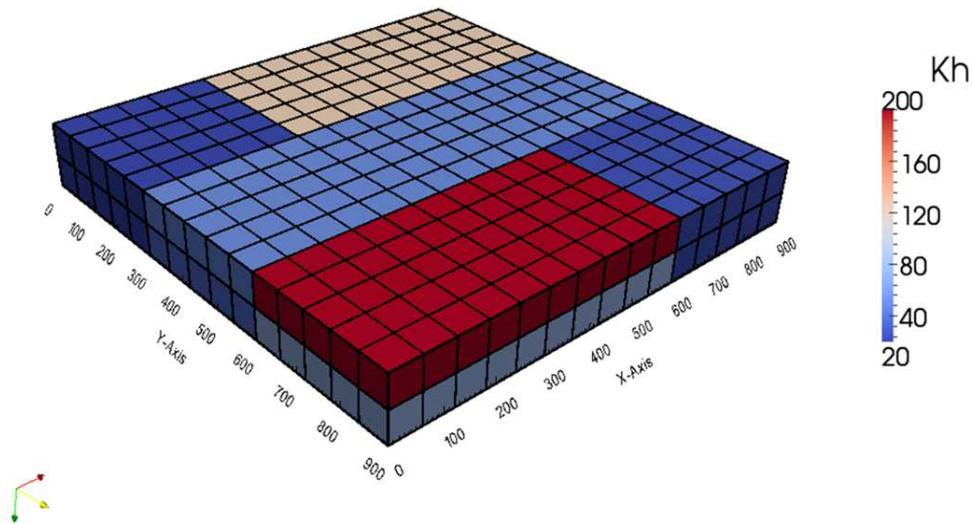


Fig. 3. Reservoir permeability distribution

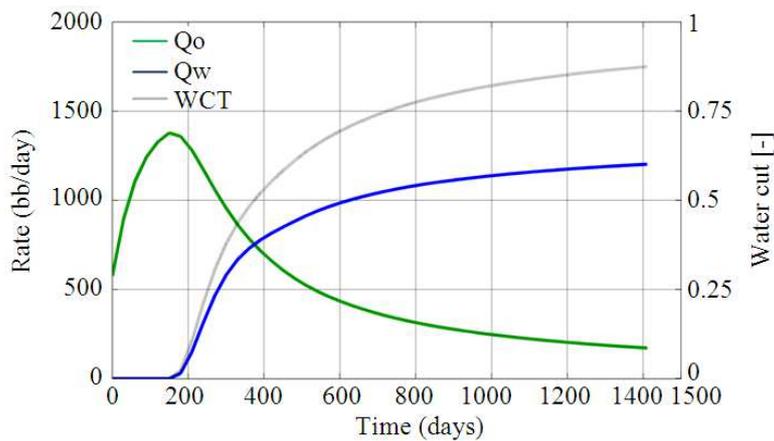


Fig. 4. Water cut, oil and water rates vs time for the production well

The wellbore radius for both wells was 0.1 m and the rock compressibility was set equal to $4.7 \cdot 10^{-5}$ L/bar.

The porous medium was initially saturated by oil with an irreducible water saturation of 0.2. Oil and water were modeled as slightly compressible fluids. The adopted compressibilities were $4.7 \cdot 10^{-4}$ L/bar and $4.0 \cdot 10^{-5}$ L/bar for oil and water, respectively.

The formation volume factor at the initial pressure was assumed equal to 1.3 bbl/STB (or m^3/scm^3) and to 1.0 bbl/STB (or m^3/scm^3) for oil and water, respectively. Fluid viscosities were assumed to be constant and equal to 0.2 cP and 0.4 cP for oil and water, respectively.

A water flooding process was simulated using an injector and a production well located at opposite corners of the reservoir. Water is typically injected into oil reservoirs to displace the oil toward the production wells. A constant water injection rate was imposed at the injection well and the corresponding pressure trend was monitored. A target liquid rate and a constraint on the minimum bottom hole pressure were imposed to the producer. The simulated period was 4 years.

The oil and water rates (also shown as water cut, WCT, which is the ratio between the water rate and the total liquid rate) and the bottom hole pressure for the production well are shown as a function of time in **Fig. 4 and 5**, respectively.

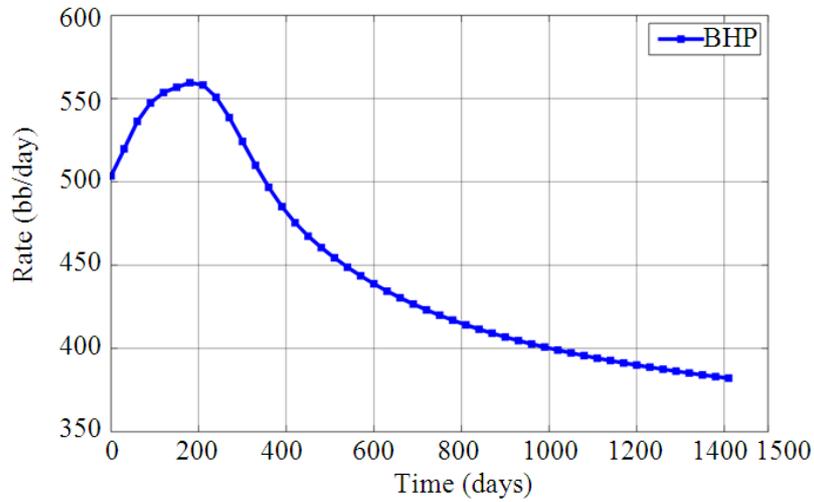


Fig. 5. Bottom hole pressure vs time for the injection well

Table 1. Mean and variance of the input parameters used during the initialization step

Parameter	Mean	σ
POR ₁	0.2	0.1
POR ₂	0.2	0.1
PERM ₁ (mD)	22.0	10
PERM ₂ (mD)	100.0	30
PERM ₃ (mD)	40.0	20
PERM ₄ (mD)	190.0	30
PERM ₅ (mD)	20.0	20
PERM ₆ (mD)	130.0	30
PERM ₇ (mD)	45.0	25
PERM ₈ (mD)	75.0	10

The goal of the application was to determine a set of “optimal” reservoir models based on the Pareto optimality criterion. The models were obtained by applying the SPEA2 algorithm to find the reservoir input parameters that successfully led to a set of objectives minimizing the difference between the simulated and historical data.

The set of input parameters are porosity and permeability (Equation 5):

$$\bar{x}^* = \{POR_1, POR_2, PERM_1, \dots, PERM_6, \} \quad (5)$$

The mean and variance of the initial population used for the synthetic case are summarized in **Table 1**. The objective functions were $F_1(x)$ and $F_2(x)$. $F_1(x)$ represents the sum of the misfit between the simulated water cut $WTC_i^*(x)$ and the historical water cut WTC_i , at the production well at each simulation time-step i (Equation 6):

$$F_1(x) = \sum_i \frac{(WTC_i^*(x) - WTC_i)^2}{\sigma_{WTC_i}^2} \quad (6)$$

The term $\sigma_{WTC_i}^2$ is the variance of the historical water cut data.

Analogously, $F_2(x)$ represents the sum of the misfit between the simulated bottom hole pressure $BHP_i^*(x)$ and the historical bottom hole pressure BHP_i at the injection well at each simulation time step i (Equation 7):

$$F_2(x) = \sum_i \frac{(BHP_i^*(x) - BHP_i)^2}{\sigma_{BHP_i}^2} \quad (7)$$

The term $\sigma_{BHP_i}^2$ is the variance of the historical bottom hole pressure data.

The SPEA2 algorithm was implemented as follows:

Step 1: Initialization. The archive size was set to 40 and the population size to 100. A total of 4000 simulations runs were launched in 40 generations. The initial population was obtained by using a normal distribution for each input parameter (**Table 1**).

Step 2: Fitness assignment. In the SPEA2 algorithm a strength value S_i is assigned to each individual of the population and of the archive. The strength value S_i is the number of solutions dominated by solution i and was calculated as in Equation 8:

$$S_i = \left| \left\{ j \in P_t + A_t \wedge i \succ j \right\} \right| \tag{8}$$

where, P_t and A_t are the population and the archive individuals at iteration t , respectively. The symbol \succ represents the Pareto dominance of i over j . Then, the raw fitness R_i of each individual was calculated using the strength value (Equation 9):

$$R_i = \sum_{j \in P_t + A_t, j \succ i} S(j) \tag{9}$$

A high value of R_i identifies an individual that is dominated by many strong (high strength value) individuals.

Then, the density information was calculated. SPEA2 uses an adaptation of the k -th nearest neighbor method (Silverman, 1986). In this application the Euclidean distance between points in the solution space was used. The k -th distance of the element i , given by σ_i^k , can be calculated as in Equation (10):

$$\sigma_i = \text{sort} \{ d_{i1}, d_{i2}, \dots, d_{iN} \} \tag{10}$$

where, σ_i^k is the k -th element of σ_i and $k = \sqrt{N_p + N_{ar}}$ with $N_p + N_{ar}$ being the number of individuals of the population and of the archive, respectively.

The density function is defined as Equation 11:

$$D_i = \frac{1}{\sigma_i^k + 2} \tag{11}$$

The classical fitness function $F(i)$ of the SPEA2 algorithm is expressed by the sum of the raw fitness $R(i)$ and the density function $D(i)$; however, an additional “social” term $L(i)$ was introduced into the fitness function (Equation 12):

$$F(i) = R(i) + D(i) + L(i) \tag{12}$$

This term allows the users to manually favor a solution that is considered acceptable even if it is not a dominant solution. $L(i)$ is proportional to the number of users who “like” the solution represented by the individual i and is expressed as in Equation 13:

$$F(i) = l_u (R(i) + D(i)) \tag{13}$$

where, l_u is the number of users that “like” the i solution.

This kind of elitism allows a user assisted search process that can lead to faster and more physically correct models.

According to the fitness function value the individuals can be divided into three categories Equation (14):

$$\begin{cases} F(i) \leq 0 & \text{"Liked" individual s} \\ 0 < F(i) < 1 & \text{Nondominated individual s} \\ F(i) \geq 1 & \text{Dominated individual s} \end{cases} \tag{14}$$

Step 3: Environmental Selection. The environmental selection is represented by the update of the archive individuals at each generation. Let N_{lk} , N_{nd} and N_{ar} be the number of “liked” individuals, non-dominated individuals and archive size, respectively; when trying to update the archive, three different situations can occur:

Case 1: $N_{nd} + N_{lk} = N_{ar}$

If the number of non-dominated and “liked” individuals fits exactly the archive size then these individuals are copied directly in the archive and the selection step is finished.

Case 2: $N_{nd} + N_{lk} < N_{ar}$

If the number of non-dominated and “liked” individuals is lower than the archive size, then all the non-dominated and “liked” individuals together with the best $N_{ar} - N_{nd} - N_{lk}$ dominated individuals are copied to the archive.

Case 3: $N_{nd} + N_{lk} > N_{ar}$

If the number of non-dominated and “liked” individuals is larger than the archive size, then an iteratively truncation process starts by removing the element among the non-dominated and “liked” individuals which has the minimum distance from another individual. The process continues until the dimension of the set is equal to the dimension of the archive.

The maximum number of “liked” solutions was also fixed (N_{lkmax}) so as to guarantee a good tradeoff between automatic and assisted exploration. When the dimension of the “liked” individuals exceeds N_{lkmax} , the process is automatically truncated using the above described mechanism:

- Step 4: Termination. The algorithm stops when the maximum number of generations is reached (40 generations). The result will be a set of individuals that are non-dominated or “liked”, located in the archive.
- Step 5: Mating selection. The mating selection is carried out by tournament selection with replacement.
- Step 6: Variation. Each input parameter is selected randomly from one of the two parents. Then a mutation operator is applied to each individual of the new population (Equation 15):

$$x^* = x + N(0, \sigma) \tag{15}$$

The mutation is represented by a normal distribution vector with mean equal to 0 and standard deviation of each parameter specified in **Table 2**.

For each case, 5 runs of the algorithm with different random seeds were carried out. The utopia point was fixed at ($f_{WTC} = 1, f_{BHP} = 0.5$).

Over the last decades, researchers have been trying to bring together the processing capacity of many remote computers to accomplish computational tasks demanding large CPUs, from cracking encryption algorithms to extraterrestrial life search. While the concept of community or distributed computing is not new, the rapidly growing number of home and office computers, along with the widespread access to the Internet, is increasing the interest in technologies able to exploit the processing power of a large number of computers located around the world.

The most relevant example of community computing is the SETI@home project, developed by the University of California in Berkley. An application of distributed computing has been used in the oil industry for real time monitoring of hydrocarbon reservoirs (Ayodelle, 2004).

At the current state of the art, it is possible to foresee a direct application of this technology to AHM, in which the simulation runs are distributed among several idle computers spread around the world using Internet.

Table 2. Standard deviation for each input parameter used during the mutation step

Parameter	Standard deviation
PERM ₁ (mD)	0.07
PERM ₂ (mD)	0.07
PERM ₃ (mD)	3.00
PERM ₄ (mD)	3.00
PERM ₅ (mD)	3.00
PERM ₆ (mD)	3.00
PERM ₇ (mD)	3.00
PERM ₈ (mD)	3.00

This approach can be implemented not only by multinational oil companies, but also by universities and research groups where computational resources are distributed in different departments.

Multi-objective optimization offers the possibility to obtain more than one solution by exploiting the Pareto optimality concept. However, it requires a larger number of evaluations of the objective function than the single-objective optimization. Since each evaluation of the objective function requires a simulation run, the process can doubtfully be carried out by a single computer.

In this study a special emphasis is put on the effective parallelization of the multi-objective optimization algorithm that takes advantage of the emerging technologies in the Web 2.0 era.

For parallelization the application was divided into five different modules, as shown in **Fig. 6**.

All the information about the projects, users, simulation results and simulation queue was stored in the database module. The working or client module is represented by a standalone application that runs the reservoir simulations received from the web module. The module, which is linked to the database, manages the work distribution among the different clients and is connected to the social module for displaying and sharing information among the users. Eventually, a developer module is available for fast prototyping and testing the optimization algorithms.

The data is stored in the database module. The main advantages of using a database are:

- The data redundancy is minimized
- A more consistent system is obtained by reducing updating errors
- The security of data is improved
- The users’ access to data is facilitated through query languages

The information contained in the database can be roughly divided into two categories: user data and simulation data. The first category contains the records of each user and the different projects currently managed by the system. The users are only allowed the execution of the simulations and the retrieval of the data from the projects for which they are authorized.

The second category contains the input and output parameters of each simulation, information about the numerical solver, information coming from the social module and details about the machine and user that run each simulation.

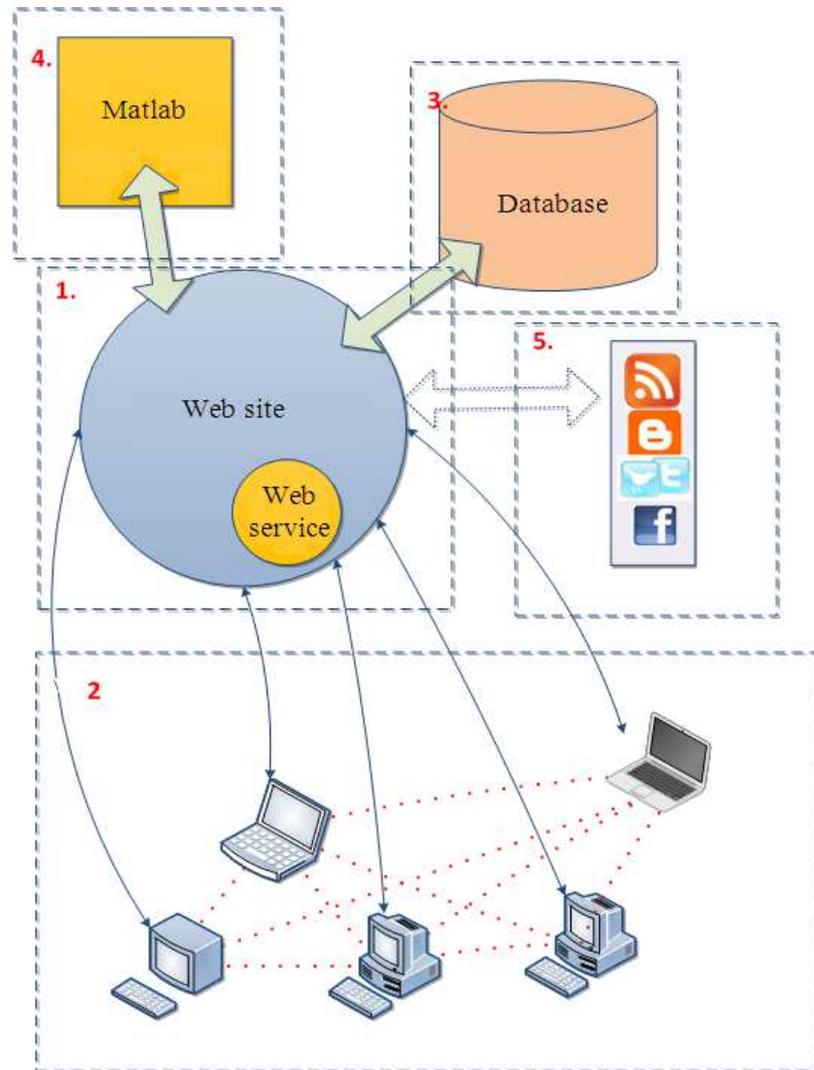


Fig. 6. Modules of the applications

The client application is a standalone application with built-in black-oil numerical simulator libraries. The clients are connected to the web module from which they receive the input data necessary to carry out the simulations. Since the calculations are CPU-intensive tasks, the clients can work only during the idle time of the CPU and can be connected and disconnected every time. A more extended version of the client application also allows 3D and 2D visualization of the simulations runs in real time.

The web module is the central module of the application; all the other modules are linked to it. Its main functions are to:

- Retrieve input simulation data and distribute it among the available working clients
- Insert new records on the database simulation queue coming from the developers' module or from the built-in optimization algorithms
- Receive information of the social plugin and insert it into the database
- Display the information of every project and simulation graphically and allow the user to interact with these items through the social plugin
- Receive the output information of the simulations from the working clients and write them in the database

The social module represents an innovative issue in the AHM field. The Facebook social plugins and a RSS feeds are implemented in the application as a proof of concept; however, in a truly shared application a more secure platform should be used.

The objective of the social module is not only to provide the user with an easier and more user-friendly history matching workflow, but also to easily interact with and to guide the optimization process. The users can select a candidate solution by using the “like” button, which in turn changes the objective function for that specific individual. The larger the number of users who confirm the preferred solution to be a good solution, the greater the possibilities of that individual participating in the mating process. As a consequence, the exploration of the solution space in the vicinity of the chosen solution is favored. The users are notified whenever substantial improvements of the objective function are obtained and whenever other users “like” a specific solution.

Two different cases were run so as to prove the efficiency of the social approach in exploring the search space close to the “liked” solutions. In the first case, a classical SPEA2 algorithm was used without “social” interaction. In the second case, the users selected a total of five individuals after ten generations; then, the algorithm continued for other thirty generations. Results of both cases are discussed in the dedicated session.

The last but not least important module is the developers module. It basically consists in an interface layer connecting Matlab® with the web module to forward input data and retrieve output data from the clients’ simulation. The incoming results are processed by a Matlab® script and new individuals are generated.

Furthermore, new algorithms can be tested in run time.

3. RESULTS

Cross-plots between different input parameters of the SPEA2 case are shown in **Fig. 7**. The individuals representing the Pareto front are depicted in red and the rest of the population in blue. Solutions exhibit a distributed Pareto front, except for the cross-plot between POR_1 and POR_2 , where a clusterization close to the base case scenario (initial porosity values 0.26 and 0.18) is evident. The matches of the water cut and bottom hole pressure are depicted in **Fig. 8 and 9**, respectively, where it is evident that almost all solutions are “good” matches. The efficiency of the algorithm of

generating solutions towards the Pareto front is pointed out in **Fig. 10**. In order to appreciate the Social Term contribution, cross-plots for two input parameters ($PERM_5$ and $PERM_8$) are shown in **Fig. 11 and 12** where no selection strategy and selection of “liked” individuals are applied, respectively. Analogous cross-plot in the solution space (water cut vs bottom hole pressure) is depicted in **Fig. 13**. The performance of the SPEA2 algorithm for the two analyzed approaches is shown in **Fig. 14**. Finally, the horizontal scalability of the algorithm is shown in **Fig. 15**. Because of the relatively limited number of processors available for this research a speed-up curve with 6 processors is shown. However, it is observed that in other realities current computational power could provide over 10 of thousands of processors.

4. DISCUSSION

The scatter of the Pareto front in the search space observed for the first analyzed case (**Fig. 7**) indicates a large diversity in the solutions, corresponding to several different configurations of the reservoir. As a consequence of the ill-posedness of the problem and non-uniqueness of solution, the presence of multiple acceptable solutions is evident in **Fig. 8 and 9**. However, in contrast with the traditional history matching approach, the SPEA2 AHM was able to provide more than one equally acceptable matches. It was verified that by introducing the social term (**Fig. 11 and 12**) a larger number of individuals is created in the neighborhood of the “liked” individuals. The same effect is shown in the solution space (**Fig. 13**): Two small clusters were obtained in the vicinity of “liked” solutions (red). The presence of these clusters proved the advantage that the social term offers in exploring solutions that would otherwise be discarded. Curves in **Fig. 14** clearly show that the contribution of the social term had a little impact on the performance of the algorithm, since the selected individuals were outside of the reference hyper-volume. The linear reduction of computational time in relation to the number of processors observed in **Fig. 15** confirms the importance of distributed computation in AHM. Although the scalability of the systems seems to be fairly good other major issues of this parallel framework, such as network latency, fault tolerance and the limitations imposed by using a centralized database need to be further investigated.

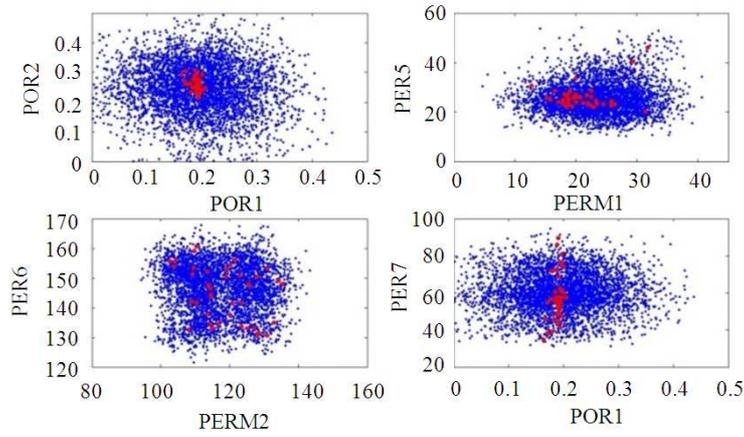


Fig. 7. Cross-plots for different input variables. Red points represent the Pareto individuals

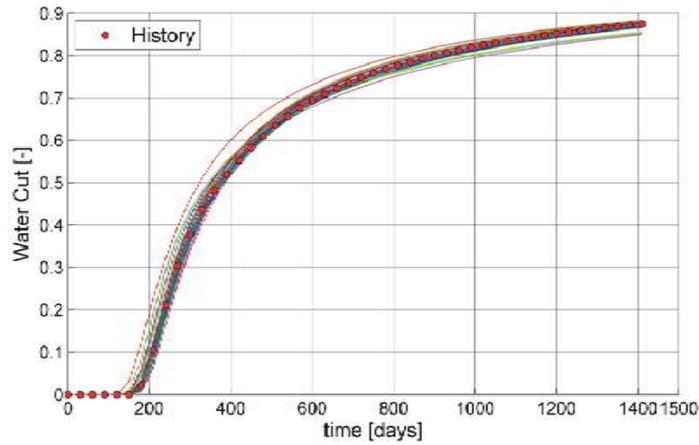


Fig. 8. Water cut at the Production well-final matching solutions

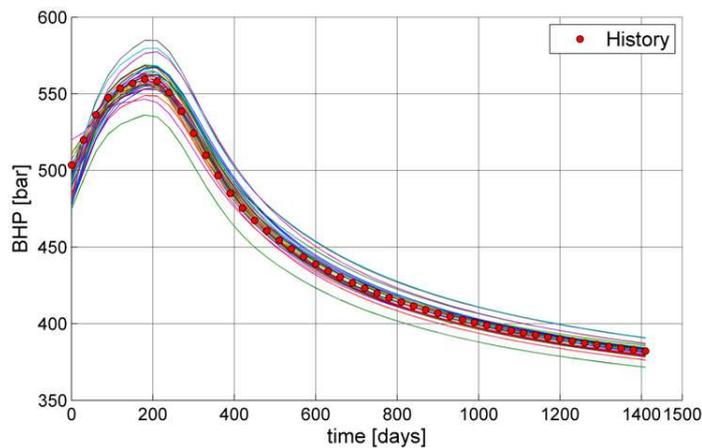


Fig. 9. Bottom hole pressure at the Injection well-final matching solutions

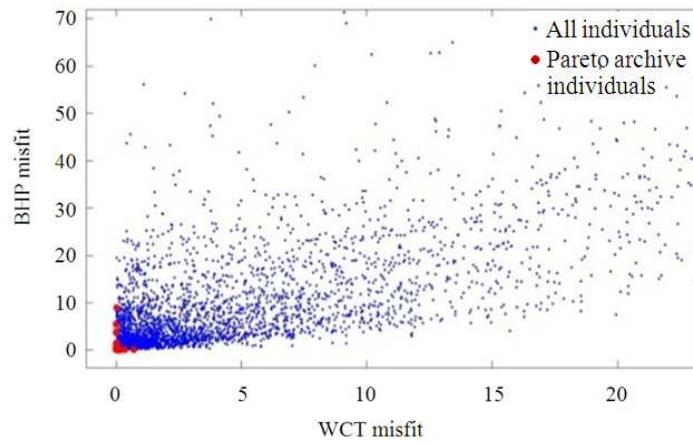


Fig. 10. Pareto front (red)

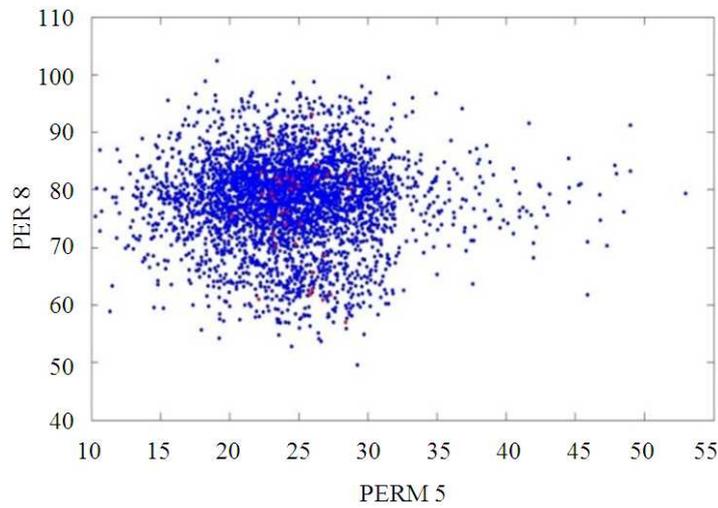


Fig. 11. Cross-plot of two permeability input parameters (perm5 and perm8) without user interaction

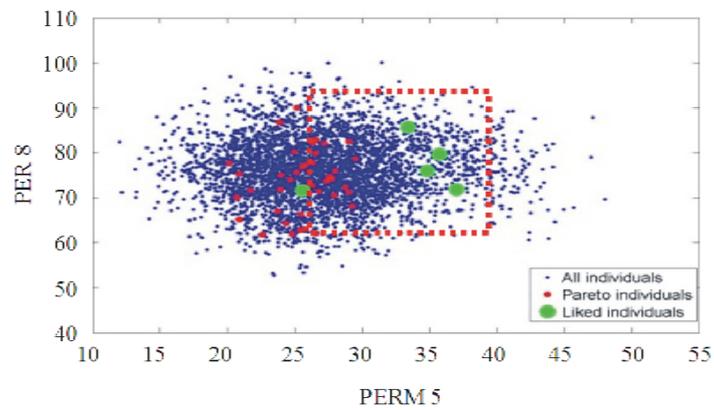


Fig. 12. Cross-plot of two permeability input parameters (perm5 and perm8) with "liked" individuals

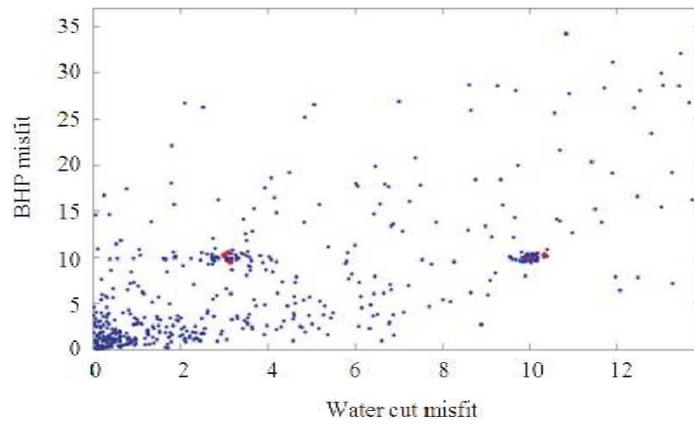


Fig. 13. Solution space for the SPEA2 with the “social” term

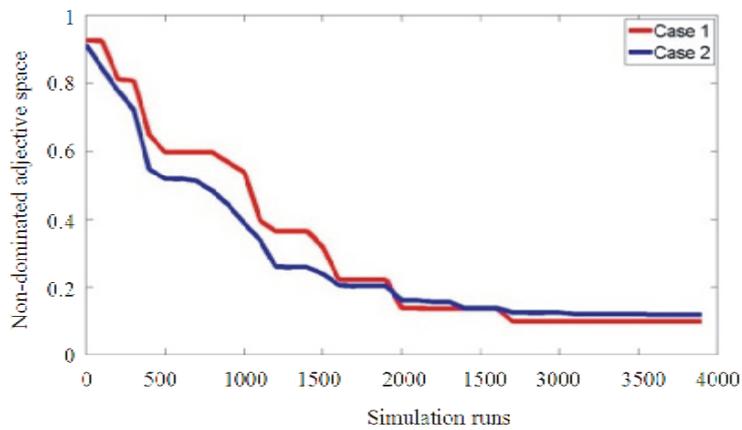


Fig. 14. Average performance values

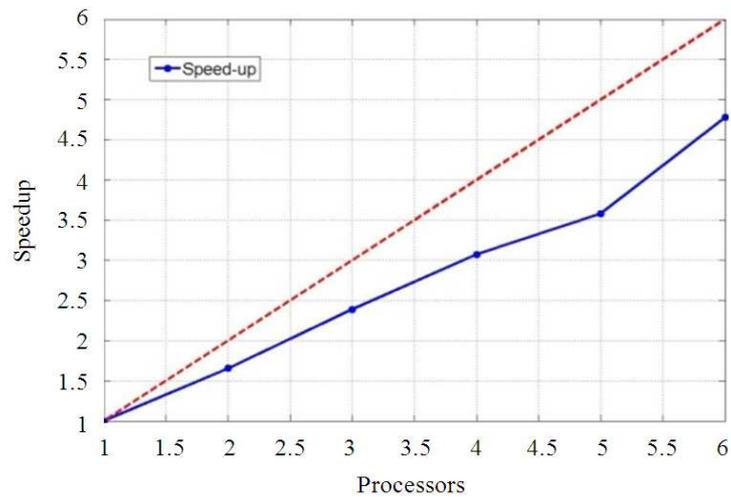


Fig. 15. Speed-up Vs number of processors

5. CONCLUSION

History matching is an indispensable and irreplaceable methodology for reservoir model calibration and for the evaluation of possible production scenarios. Assisted history matching is an emerging methodology, which is very attractive because it can facilitate, improve and speed-up the history matching process, yet providing multiple solutions.

In this study, an innovative framework for the assisted calibration of reservoir numerical models was presented. The study was triggered by the need for an algorithm with the capability of taking advantage of a large network of computers and users to process the significant number of simulations required by multi-objective optimization algorithms.

A multi-objective evolutionary algorithm (the modified Strength Pareto Evolutionary Algorithm named SPEA2) was developed for the calibration of reservoir numerical models. A “social” contribution was added to the fitness function of the SPEA2 algorithm to facilitate the exploration of the solution space near to the selected individuals. The developed algorithm was coupled to a collective computational network.

The reservoir selected to test the effectiveness of the proposed methodology was a heterogeneous and anisotropic oil-bearing formation with a simple geometry. Water flooding through an injector well was simulated to displace the oil towards the production well. The model was characterized by a total of ten parameters, namely two porosity regions and eight permeability regions. These were changed to find the Pareto front in the SPEA2. The results showed that the SPEA2 algorithm found a representative portion of the Pareto front, in which a set of solutions matched the base or historical data. However, equally acceptable matches could be found with different input parameters. This highlighted the advantage of using an evolutionary algorithm to find a representative set of matches, as opposed to the single non-unique solution that is obtained from manual calibration.

The web module provided an easy way to interact with the algorithm and search mechanism through the SPEA2 objective function. Additionally, the possibility for any competent user to steer the selection of the fittest individuals (“like” option) was introduced. The results showed the horizontal scalability of the method and demonstrated the importance of the parallelization for speeding up the overall process. However, more work is needed in order to guarantee the security of the information in social networks.

Future work includes testing the efficiency of the system by modeling the expert opinion stochastically or deploying the system on a large network of reservoir engineers.

6. REFERENCES

- Ayodelle, O.R., 2004. Distributed computing for real-time petroleum reservoir monitoring. *J. Canadian Petroleum Technol.*, 43: 9-12. DOI: 10.2118/04-05-TN1
- Cancelliere, M. and F. Verga, 2012. Simulation of unconventional well tests with the finite volume method. *Petroleum Scie.*, 9: 317-329. DOI: 10.1007/s12182-001-0215-6
- Cullik, A.S., D. Johnson and G. Shi, 2006. Improved and more rapid history matching with a nonlinear proxy and global optimization. *Proceedings of the SPE Annual Technical Conference and Exhibition*, Sep. 24-27, San Antonio, Texas, USA. DOI: 10.2118/101933-MS
- Fokker, P.A., J. Renner and F. Verga, 2013. Numerical modeling of periodic pumping tests in wells penetrating a heterogeneous aquifer. *Am. J. Environ. Sci.*, 9: 1-13. DOI: 10.3844/ajessp.2013.1.13
- Gomez, S., O. Gosselin and J.W. Barker, 2001. Gradient-based history matching with a global optimization method. *SPE J.*, 6: 200-208. DOI: 10.2118/71307-PA
- Landa, J., R.K. Kalia, A. Nakano, K. Nomura and P. Vashishta, 2005. History match and associated forecast uncertainty analysis-practical approaches using cluster computing. *Proceedings of the International Petroleum Conference*, Nov. 21-23, Doha, Qatar. DOI: 10.2523/10751-MS
- Mohamed, L., M. Christie and V. Demyanov, 2011. History matching and uncertainty quantification: multiobjective particle swarm optimisation approach. *Proceedings of the SPE EUROPEC/EAGE Annual Conference and Exhibition*, May 23-26, Vienna, Austria. DOI: 10.2118/143067
- Oliveira, V.C.D., A.D.A. Mota and L.T. Moreira Mota, 2013. Impacts of application usage and local hardware on the throughput of computer networks with desktop virtualization. *Am. J. Applied Sci.*, 10: 117-122. DOI: 10.3844/ajessp.2013.117.122
- Oliver, D.S. and Y. Chen, 2010. Recent progress on reservoir history matching: A review. *Comput. Geosci.*, 15: 185-221. DOI: 10.1007/s10596-010-9194-2

- Riegert, R.K., J.K. Axmann, O. Haase, D.T. Rian and Y-L. You, 2001. Optimization methods for history matching of complex reservoir. Proceedings of the SPE Reservoir Simulation Symposium, Feb. 11-14, Houston, Texas. DOI: 10.2118/66393-MS
- Schaaf, T., B. Coureaud and N. Labat, 2008. Using experimental designs, assisted history matching tools and bayesian framework to get probabilistic production forecasts. Proceedings of the Europec/EAGE Conference and Exhibition, Jun. 9-12, Rome, Italy. DOI: 10.2118/113498-MS
- Selberg, S., B.E. Ludvigsen, A. Diab and T. Harneshaug, 2006. New era of history matching and probabilistic forecasting-a case study. Proceedings of the SPE Annual Technical Conference and Exhibition, Sep. 24-27, San Antonio, Texas, USA. DOI: 10.2118/102349-MS
- Silverman, B.W., 1986. Density Estimation for Statistics and Data Analysis. 1st Edn., Chapman and Hall, London, ISBN-10: 9780412246203, pp: 175.
- Verga, F., M. Cancelliere, D. Viberti, 2013. Improved application of assisted history matching techniques. J. Petroleum Sci. Eng. DOI: 10.1016/j.petrol.2013.04.021
- Wolpert, D.H. and W.G. Macready, 1997. No free lunch theorems for optimization. IEEE Trans. Evol. Comput., 1: 67-82. DOI: 10.1109/4235.585893
- Zitzler, E., 1999. Evolutionary algorithms for multiobjective optimization: Methods and applications. PhD Thesis, Swiss Federal Institute of Technology, Zurich.
- Zitzler, E., M. Laumanns, L. Thiele, 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-Report 103. DOI: 10.3929/ethz-a-004284029