ARTICLE IN PRESS

Journal of Petroleum Science and Engineering (xxxx) xxxx–xxxx

Contents lists available at ScienceDirect

Journal of Petroleum Science and Engineering

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

Reducing the computation time of well placement optimisation problems using self-adaptive metamodelling

Mohammad Sayyafzadeh

The University Of Adelaide, SA 5005 Australia

ARTICLE INFO

Keywords:
Surrogate
Genetic algorithm
Artificial intelligence
Model management
Well placement optimisation

ABSTRACT

A group of optimisers, widely used for well placement problems, are population-based algorithms. These algorithms have a major downside; they are computationally demanding. Online metamodelling is a CPU-time reduction technique in which the exact fitness function (EF) is substituted, partially, with an approximation function (AF), often known as proxy or surrogate. Particular attention should be paid to this replacement, as it may cause convergence to an arbitrary optimum solution. A successful model management strategy can prevent the possible misdirections, by applying the EF effectively, during the optimisation process. Designing such a strategy is an active area of research in different disciplines. The main motivation behind this study was to develop a self-adaptive model management strategy for surrogate-assisted algorithms in such a way that the same quality results, as those obtained by the corresponding unassisted (typical) algorithm, are delivered with less computation.

In the proposed model management strategy, two surrogates are utilised. The first surrogate approximates the fitness function landscape, and the second one estimates the fidelity (accuracy) of the first surrogate over the search space. According to the estimated fidelity, the probability of using the EF is calculated for each individual, and then the algorithm stochastically decides to use the EF or AF. A heuristic fuzzy rule defines the range of probabilities in each evolution-cycle, based on the average fidelity of the second surrogate. The strategy was implemented on a genetic algorithm, with two neural networks, as the surrogates. The robustness of the proposed online-learning algorithm was analysed using a benchmarking analytical function and a semi-synthetic reservoir model, PUNQ-S3. The outcomes were compared with the results achieved by three algorithms, an unassisted algorithm, an offline-learning surrogate-assisted algorithm, and an online-learning surrogate-assisted algorithm with a random selection model management strategy. The comparison showed that the online-learning algorithm with the proposed strategy can outperform the other algorithms.

1. Introduction

The locations of wells (producers and injectors) in a reservoir have significant impacts on hydrocarbon recovery factor and accordingly the revenue from the asset. Therefore, finding an optimal scenario for the placement of the wells is a crucial task. In order to optimise well placement, several elements should be taken into consideration, such as hydrocarbon-in-place, reservoir connectivity, fluid and petrophysical properties, operational/drilling costs and constrains, etc. (Bouzarkouna et al., 2012). Reservoir simulation is a tool applied to obtain insights about the reservoir response in respect to development strategies. Thus, using a reservoir simulator, it is possible to assess the efficiency of different plans and perform a sensitivity analysis.

In order to find an optimal well placement scenario, the following procedure, typically, is used (Beckner and Song, 1995). First, the decision variables (DVs) are defined. These are the parameter sought to be optimised, for instance wells' location and trajectory. This step defines the search space. Second, constraints are implemented which creates a feasible region in the search space. The solution candidates (scenarios) are the members of this subspace (solution-space). Third, an objective (fitness) function is formulated to measure the goodness of the solution candidates and differentiate them. Net Present Value (NPV) is usually applied, as the objective function (Beckner and Song, 1995; Guyaguler and Horne, 2001). Finally, an optimisation algorithm is used to search the feasible space to find the best solution.

In order to evaluate the fitness of each solution candidate, a reservoir simulation should be executed. Due to the computational intensity of the numerical simulation, the nonlinearity of the fitness function and potential high-dimensionality of the search space, a robust optimisation algorithm is required to find the maximum of the objective function with the minimum CPU-time. Thus far, several algorithms have been applied. Zandvliet et al. (2008), Wang et al.

E-mail address: mohammad.sayyafzadeh@adelaide.edu.au.

http://dx.doi.org/10.1016/j.petrol.2016.12.015
Received 28 July 2016; Received in revised form 6 November 2016; Accepted 12 December 2016

Nomenclature

a	Minimum probability
b, b'	ANN's parameter
CF	Cash flow
DR	Discount rate
d_{cal}	Simulation data required for NPV calculation
$f(M,x)$	Exact fitness function
$\hat{f}(b, w, v, x)$	Approximation fitness function
H	Evolution-control size
$h(t)$	Activation function
L_1 & L_2	Normalising value
M	Simulation-model
N	Number of samples or the number of EF-calls
N_x	The dimension of X
N_y	The dimension of Y
P,Q,R	The number of neurons
p_2	Probability calculated based on the average accuracy of the second surrogate
p_i	Probability of selecting EF for individual x_i
q	Production rates
R_i	Normalised approximated relative difference
T	Years of production for calculation of NPV
$u(\vec{x})$	Average relative difference
v	ANN's structure
w	ANN's parameter
X	Search-space
x_i	An element of $\vec{x}$
$\vec{x}$	An individual (a vector in X)
Y	Fitness space
y_i	Exact fitness value corresponds to x_i (a vector in Y)
$Z(x_i)$	Relative difference for individual x_i
$\hat{Z}(b', w', v', x_i)$	Approximated Relative difference for individual x_i

(2007), Forouzanfar et al. (2010) and Zhang et al. (2010) proposed the use of adjoint-based algorithms. This type of algorithms can offer a significant CPU-time reduction in comparison with heuristic, meta-heuristic and stochastic algorithms. But, they may get stuck in a local minimum (Onwunalu and Durlofsky, 2010). In stochastic algorithms, the search space is visited randomly to escape from the local minima. This procedure is the basis for many stochastic optimisers, such as Simulated Annealing (SA) (Beckner and Song, 1995) and Simultaneous Perturbation Stochastic Approximation (SPSA) (Bangerth et al., 2006). Bangerth et al. (2006) compared SPSA with SA, and it was observed that SPSA is more computationally efficient. Onwunalu and Durlofsky (2010) pointed out two drawbacks of SPSA, finding a proper step size and the gradient-based nature, and as an alternative, employed a population-based algorithm.

In population-based algorithms, instead of modifying a single solution candidate, a set of candidates, known as population, is being modified in each iteration, known as generation. With the recent advances in computer hardware, these algorithms are becoming more attractive, due to the fact that most of them can be implemented on clusters. Genetic Algorithm (GA) has a relatively long history in production optimisation and history-matching problems. The following are some of the studies in which a GA was applied for (Montes et al., 2001; Bittencourt and Horne, 1997; Salmachi et al., 2013; Emerick et al., 2009; Mamghaderi et al., 2013; Sayyafzadeh et al., 2012; Sayyafzadeh and Keshavarz, 2016). The other population-based algorithms used for well placement optimisation problem are covariance matrix adaptation evolutionary strategy (Bouzarkouna et al., 2012), particle swarm (Onwunalu and Durlofsky, 2010; Siavashi et al., 2016) and differential evolution (Nwankwor et al., 2013).

Although the population-based algorithms may outperform the others, they have a major downside. These algorithms are computationally expensive (Filho and Gomide, 2006), as they require many fitness function calls during the optimisation process. Reduced-order modelling is a technique for the reduction of flow simulation time and was applied for well control optimisation problems (He and Durlofsky, 2014; Jansen and Durlofsky, 2016; Cardoso et al., 2009). But, in this study, the aim is to reduce the number of (full-order) simulations, using mathematical surrogates.

Surrogate-assisted algorithms are widespread techniques, for CPU-time reduction, in which the exact fitness function (EF) is partially (if online-learning) or completely (if offline-learning) substituted by an approximation function (AF), often called surrogate, metamodel or proxy. A surrogate is trained (tuned) by a set of samples taken from the original fitness function landscape. The following papers are examples in which a surrogate-assisted evolutionary algorithm was used for production and/or well placement optimisation (Zubarev, 2009; Tupac et al., 2007; Guyaguler et al., 2000). Most of the applied surrogate-assisted algorithms in Petroleum discipline literature used an offline-learning scheme in which i. An experimental design method is utilised to define the sample sites, and then, ii. By computing the fitness value over the sites with the reservoir simulator, the sample set is generated and used for training a surrogate, and finally, iii. The surrogate is solely used to estimate the fitness of every proposed individual throughout the optimisation process. Zubarev (2009) mentioned that such a practice may misdirect the optimisation to an arbitrary optimum; especially for problems with complexity (Bouzarkouna et al., 2012). This is in line with other disciplines' literature (Razavi et al., 2012; Jin, 2005, 2011).

Surrogates might not have the capabilities to approximate the global optimum, but, they can provide an overview of the entire fitness function landscape and good estimation around the sampled regions, Fig. 1. It has become a normal practice, particularly in other disciplines, to apply the AF in conjunction with the EF for the fitness evaluation, known as online-learning (Razavi et al., 2012; Jin, 2011). To use the EF effectively and efficiently through the optimisation process, a model management (evolution-control) strategy should be used. Thus far, various strategies have been introduced (Jin, 2005, 2011; Jin et al., 2001). In an online-learning scheme, to evaluate each of the proposed individual in every population, the algorithm decides, based on the implemented model management strategy, to choose between the two measures (EF and AF), and meanwhile the AF is retrained (sequentially when a few more individuals evaluated by the EF are added to the sample pool). Some of the common model management strategies are reviewed in the next section, and the pros and cons of them are discussed. In the following studies, a similar concept was used for well placement optimisation problems. Bouzarkouna et al. (2012) utilised a metamodel as a local search tool in a CMA-ES algorithm and observed a fair CPU-time reduction. In their algorithm, the full capacity of the surrogate may not be used, since the global search is conducted mainly by EF measurements. Guyaguler et al. (2000) used a hybrid genetic algorithm (polytope) with proxies. In their algorithm, the search (new population generation) is conducted, based on AF measurements, and only the best-ever solution is re-evaluated by the EF. Yeten et al. (2002) used an artificial neural network as the proxy. In their algorithm, the criterion for selecting the individual (to be re-evaluated by EF) is based on the fitness value measured by the AF, and hence the search is conducted mostly by the AF. In the last two techniques, a pre-mature convergence may occur (Jin, 2005), e.g., in Fig. 1, the true solution is in right hand side, while the AF estimates a low fitness value for that region, in such a case, the EF does not get the chance to evaluate any individual in this region and those sections remain un-sampled. Researchers suggested to use a