Optimization of a reservoir development plan using a parallel genetic algorithm

Jonathan N. Carter and John D. Matthews
Department of Earth Science and Engineering, Imperial College, London SW7 2AS, UK (e-mail: j.n.carter@imperial.ac.uk)

ABSTRACT: A parallel genetic algorithm has been applied successfully to design a production plan that is substantially superior to that obtained using a conventional engineering approach. The reservoir, a dipping structure, was expected to yield optimum production using a rolling line drive from downdip to updip positions. The simulation allowed for 3800 positions for each of 11 wells, giving a total of 1.3×10^{31} options. The genetic algorithm sampled 1650 of these and was able to identify seven solutions that would increase production by over 30% compared with the rolling line drive. In contrast, a random search using 850 samples managed to find only two plans that improved production; in each of these cases the improvement was less than 1%.

KEYWORDS: *oil production, shallow marine, faults, optimization*

INTRODUCTION

The optimization of a production plan can make a major difference to the economic viability of a petroleum reservoir. Reservoir management (Thakur 1996) is the process of choosing an optimal development plan for a reservoir, which adds maximum value to a company's portfolio. The process of finding the optimum development plan is rarely formalized. A few well positions are chosen on the basis of standard reservoir engineering rules governing areal and vertical sweep (Craig 1971), and then a simulator is used to refine these positions. This process is acceptable only if the simulation model is an accurate representation of the reservoir. Since this cannot be guaranteed, the reservoir engineer will often choose final well positions for the development plan which are not those recommended by the simulations.

Operational research methodologies can be used to improve this process. There are two areas that need to be addressed: formulating efficient methodologies to define an objective function that captures the risks and opportunities of a proposed development plan, and search algorithms that can efficiently identify development plans that will be optimal using the appropriate objective function.

This paper describes a robust and flexible optimizer that can tackle the difficult optimization problem of development plan selection. Its application is illustrated on a realistic test problem, of a reservoir containing $60\,830 \times 10^3$ m^3 of oil modelled with 78 720 active grid-blocks in a black oil simulator. The properties of the resulting development plan are discussed. The test reservoir is drawn from the SAIGUP suite of faulted shallow-marine reservoirs (Manzocchi *et al.* 2008*a*). The next section describes the general problem and the specific test problem that the study attempts to solve. This is followed by a detailed description of the genetic algorithm used. Finally, the article considers the effectiveness of the optimization process and the value of the best development plan proposed during the optimization.

OPTIMIZATION OF RESERVOIR PRODUCTION PLANS

Before applying a formal optimization method to the problem of identifying an optimal development plan, four elements have to be considered.

- **The parameters**: these are all the variables within the development plan that can be adjusted, and which are to be considered as part of the problem. Depending on the situation these might include: the number, type and location of wells; the pressures and rates at which wells operate; and the timing of events, such as the drilling schedule.
- **The model**: the role of the model is to take a development plan as specified by particular values of the variables and return a set of numbers that can be used to evaluate the objective function. The model must be able to resolve any conflicts that exist with the specified development plan; it should also handle any uncertainties. This may mean that the model has stochastic elements, and may consist of several sub-models. If the development plan is unfeasible, then the model should return a dummy set of numbers that can still be evaluated by the objective function.
- **The objective function**: the objective function will take all of the information generated by the model and turn it into a single number. This number can then be compared with the equivalent number produced by a different set of variable values, and the best set of variables selected. If there are many conflicting objectives it may be difficult to produce a single measure of optimality (Deb 2001).
- **The optimizer**: a number of issues need to be considered when making the choice of optimizer. Does the optimization algorithm require gradients and does the model provide them? How easy is it to use the algorithm and get it to work effectively? Is the algorithm efficient and does it produce one 'best' answer or a suite of answers? Can it handle noise, and uncertainty, in the objective function? Is it robust to problems within the model? Can it handle

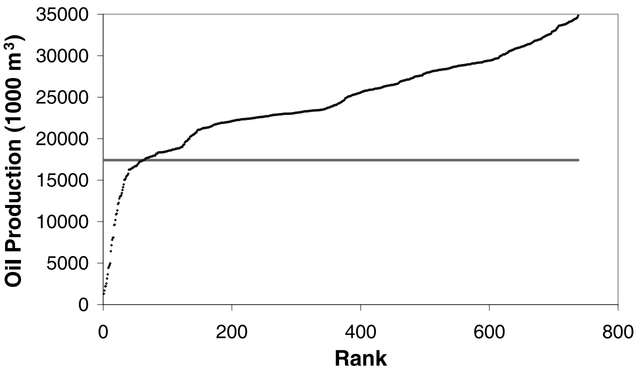

Fig. 1. The distribution of the total oil production from each of 738 sample reservoirs models.

multi-modal problems? Can it be parallelized easily, if required?

There have been a number of previous attempts at applying formal optimization methods to well placement problems (Aanonsen *et al.* 1995; Beckner & Song 1995; Bittencourt & Horne 1997; Centilmen *et al.* 1999; Guyaguler & Gumrah 1999; Guyaguler *et al.* 2002; Pan & Horne 1998; Rogers & Dowla 1994). The general aim of most of these previous attempts has been to reduce the computing effort needed. Various techniques have been tried with this as the objective: reducing the scope of the problem; using simpler models; reduced physics simulators; or proxy methods. The effect of these methods is either to reduce the quality of the resulting function evaluations, and/or to complicate the process. Guyaguler *et al.* (2002) demonstrated the complexity of the well placement problem. They showed, using exhaustive search, that for a single well placement problem the response surface is very noisy, with at least 20 local optima surrounding the global optima. Gradient-based methods are sure to fail on this problem.

THE TEST PROBLEM

The SAIGUP project (Manzocchi *et al.* 2008*a*) involved the creation of 400 sedimentological models, which, when faulted in different ways, generated approximately 12 000 synthetic clastic reservoirs (Howell *et al.* 2008; Manzocchi *et al.* 2008*b*). Reservoir production from these models was then simulated using a range of development plans (Matthews *et al.* 2008). Figure 1 shows the production when a particular development plan is applied to nine sedimentological models, each in an unfaulted state and faulted by nine separate fault systems, each characterized by nine different fault-rock permeability models. The nine selected sedimentologies were chosen to be representative of the full set of models and are illustrated in Manzocchi *et al.* (2008*a*). A typical original oil-in-place is about $60\,000 \times 10^3$ m^3, with the production representing a recovery factor of 30–60%. In many cases the production is quite reasonable. However, about 5% of the reservoirs have poor production, and the cause appears to be poor fault transmissibility combined with compartmentalizing fault systems (Manzocchi *et al.* 2008*a*, *b*).

The aim of the test problem was to design a development plan for one of the poor producing reservoirs identified in Figure 1. The reservoir chosen produced $17\,415 \times 10^3$ m^3 over the thirty-year production life studied, using the standard development plan from within the SAIGUP project (Manzocchi *et al.* 2008*a*). This amount of oil production is represented by the horizontal line in Figure 1. The variables for the test problem were the x and y locations of each of the eight producers and three injectors used in the original development plan. Given that the reservoir had poor fault transmissibility, one could have used horizontal wells to improve production. However, in order to compare the results directly with other cases within the SAIGUP project, it was decided to use only vertical wells completed over the full height of the reservoir. In a future study these constraints could be relaxed.

In total there were 22 variables, with each well capable of being placed at any one of 3800 locations. This gives approximately 1.3×10^{31} different development plans to choose from. The objective function was to maximize the total oil production at the end of thirty years. In a real study one would expect the objective function to be based on an economic evaluation of the proposed development plan. The reservoir model grid was of size $(40 \times 120 \times 20)$ 96 000 grid-blocks, of which 78 720 were active grid-blocks. The production was simulated using the MORE simulator (Young & Hemanth-Kumar 1991), supplied to the SAIGUP project by Roxar Ltd.

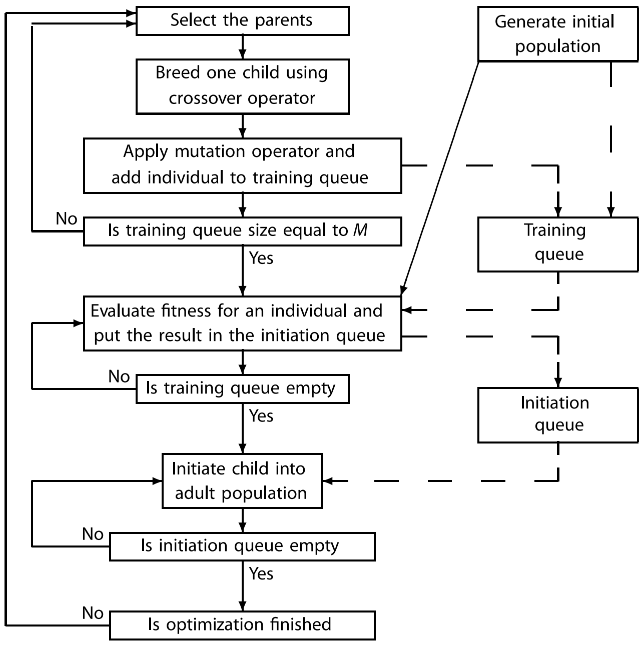

Fig. 2. The general structure of a basic genetic algorithm.

GENETIC ALGORITHM

The structure of a basic serial genetic algorithm (GA) is shown in Figure 2. The starting point of the algorithm is the generation of a sequence of possible solutions to the problem. Each solution is known as an individual, with all of the individuals forming a population. These individuals are solutions to the problem only in the sense that they represent a way of placing the wells into the reservoir. They are not expected to be optimal and they are usually generated randomly. For each of the individuals the objective function is then evaluated; for this problem the objective function was the total oil produced after thirty years. The individuals can now be put into the adult population. This is then followed by the breeding of new individuals by using selection, crossover and mutation. These new individuals have their objective functions evaluated and may then enter the adult population. This process repeats itself until the optimization is stopped. Each of the steps can be done in more than one way. For a more detailed introduction to the genetic algorithm method, the reader is referred to one of the many introductory texts, such as Carter (2003) or Mitchell (1998).

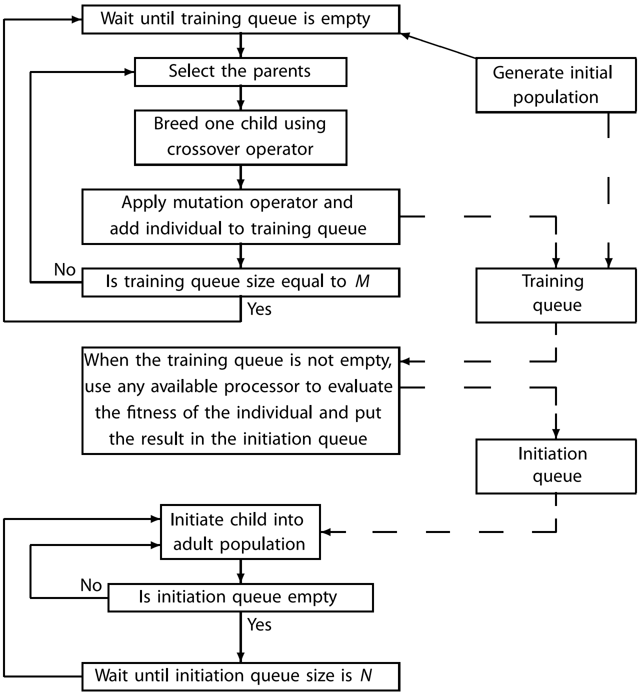

Fig. 3. The general structure of the parallel genetic algorithm from this study.

In Figure 2 there are two queues, the training queue and the initiation queue, which are not normally mentioned in introductions to the GA methodology. The training queue is the list of individuals that have been generated but have not yet had their objective function evaluated. The initiation queue is the list of individuals, and their objection function value, that have not yet been placed into the adult population. These queues have been named after the cultural idea that children should undergo a period of training before they take part in an initiation ceremony allowing them to become part of the adult population. The size that the queues are allowed to reach depends on the exact implementation of the GA. In a generational replacement scheme, both queues have a maximum size equal to the population size. In a steady-state scheme the queue sizes are both one.

Figure 3 shows how the serial GA scheme is adapted for our multi-processor implementation. The main change is that where, in Figure 2, there was a single control loop, in Figure 3 this has become three separate control loops that communicate via the two queues. The first control loop generates individuals whenever the training queue becomes empty and places them in the training queue. The second control loop takes individuals from the training queue, evaluates their objective functions and places the result in the initiation queue. The final control loop takes individuals from the initiation queue and places them into the adult population. No detailed analysis to determine the optimal lengths for the two queues has been carried out. Based on previous experience the length of the training queue, M, was set to five and the length of the initiation queue, N, to three.

The reason for this set up is the variability in the time required to evaluate the objective function for an individual. Each evaluation of the objective function for an individual requires a complex simulation of a reservoir. The time required for this varied between two minutes and forty hours. The average was four hours, with 80% taking between two hours and ten hours. If one was to use the structure of a serial GA, as shown in Figure 2, but evaluate the objective functions in parallel, it is likely that significant amounts of computer time would be wasted whilst waiting for a few simulations to finish.

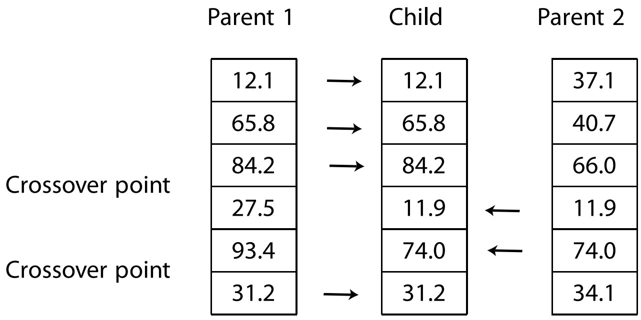

Fig. 4. Example of a two-point crossover acting on the chromosome of two parents to produce a single child.

The details of how individuals are generated from the adult population and how they are initiated into the adult population are exactly the same as for a serial GA. A number of schemes were used and all those selected for this work are very simple. The performance of the whole algorithm could be expected to improve if other choices were made (this is the subject of ongoing research). For a review of possible schemes for real variable GAs, see Ballester & Carter (2003).

Selection scheme

A two-person tournament (Goldberg & Deb 1991) has been used to select each parent from an adult population of 25 individuals. To select a parent, first choose randomly two different individuals from the parent/adult population. These two individuals then compete for the right to be a parent, with the fitter individual, i.e. the individual with the better objective function, always winning. Parents return to the pool of potential parents before the next selection process is started.

Crossover operators

The chromosome, i.e. the list of variables, used is a vector of real numbers. Two crossover operators were used to generate children, with a 50% probability of selection for each. The first was a simple two-point crossover, as illustrated in Figure 4, with the crossover points being chosen at random. The second was the SBX operator (Deb & Agrawal 1995; Deb & Bayer 1999), which operates on a gene-by-gene basis, a gene being a single variable in this problem; the genes of an offspring are biased towards those of one of its parents. It has one user-defined parameter that controls just how closely related to its parent is the offspring. The values of the offspring variables $y_i^{(1)}$ and $y_i^{(2)}$ are given by

$$y_i^{(1)} = 0.5((1 + \beta_q)x_i^{(1)} + (1 - \beta_q)x_i^{(2)}) \tag{1}$$

$$y_i^{(2)} = 0.5((1 - \beta_q)x_i^{(1)} + (1 + \beta_q)x_i^{(2)}) \tag{2}$$

where β_q is given by

$$\beta_q = \begin{cases} (2u)^{\frac{1}{\eta+1}} & 0.0<u\leq 0.5 \\ \left(\dfrac{1}{2(1-u)}\right)^{\frac{1}{\eta+1}} & 0.5<u<1.0 \end{cases} \tag{3}$$

and u is a random number with $u \in (0,1)$, η is a non-negative real number set by the user (1.0 in this work) and $x_i^{(1)}$ and $x_i^{(2)}$ are the variables from the parents.

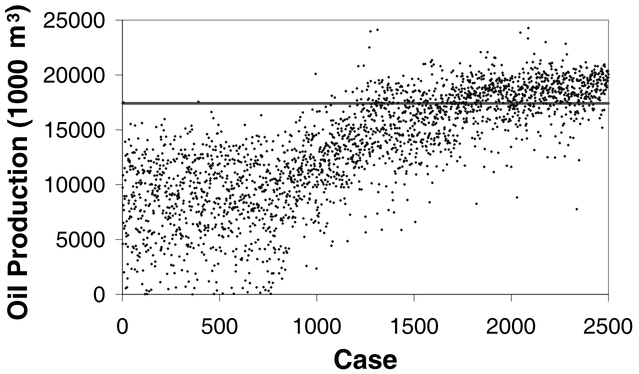

Fig. 5. Production achieved in 2500 simulations by the two optimizers. The solid line is the value achieved by the hand-designed solution (Plan-1). The first 850 cases were generated randomly; the last 1650 cases were generated by the genetic algorithm.

Mutation scheme

Because the SBX operator naturally introduces some variation at the level of individual genes (vector components), it was not thought necessary to have an explicit mutation operator.

Initiation/culling scheme

A child is transferred from the initiation queue to the adult population using random replacement. One of the current adults is randomly selected, without reference to its fitness, and replaced by the incoming child. This has the advantage that the adult population does not become dominated by the descendants of a single individual and a high level of diversity is maintained. The disadvantage is that a high quality individual can disappear from the adult population, before that individual has influenced the population.

THE RESULTS

The results will be considered in two ways: first, the performance of the GA as an optimizer will be scrutinized; secondly, the performance of the proposed development plan will be considered from a reservoir engineering viewpoint.

Optimization performance

Figure 5 shows the objective function versus the number of simulations (individuals) evaluated. In total 2500 development plans were tested, which took approximately 19 days using a cluster of 24 SUN Ultra 5 workstations. The first 850 cases used a simple random search to give a baseline against which to judge the GA. The range of productions is $(0, 17\,560 \times 10^3)$ m^3, with a mean of 8582×10^3 m^3. The next 1650 cases were generated using the GA described; the range of productions is $(2350 \times 10^3, 24\,276 \times 10^3)$ m^3, with a mean of $16\,038 \times 10^3$ m^3. The final population of 25 individuals has a range of $(16\,368 \times 10^3, 20\,859 \times 10^3)$ m^3, with a mean of $19\,314 \times 10^3$ m^3. The best three cases were: case 2088 with value $24\,276 \times 10^3$ m^3, case 1312 with value $24\,177 \times 10^3$ m^3 and case 1276 with value $23\,964 \times 10^3$ m^3.

From these results one can see that the GA is successfully searching and doing better than the random search. The base-case development plan, designed by hand, has a production of $17\,415 \times 10^3$ m^3. During the random search this was bettered on only two occasions, while the GA produced seven development plans that improved the production by more than 30%. It is not possible to draw any stronger conclusions about the performance of the GA. Due to its stochastic elements one should run multiple examples to test its performance. Also one should use a better optimization algorithm than random search for the comparison.

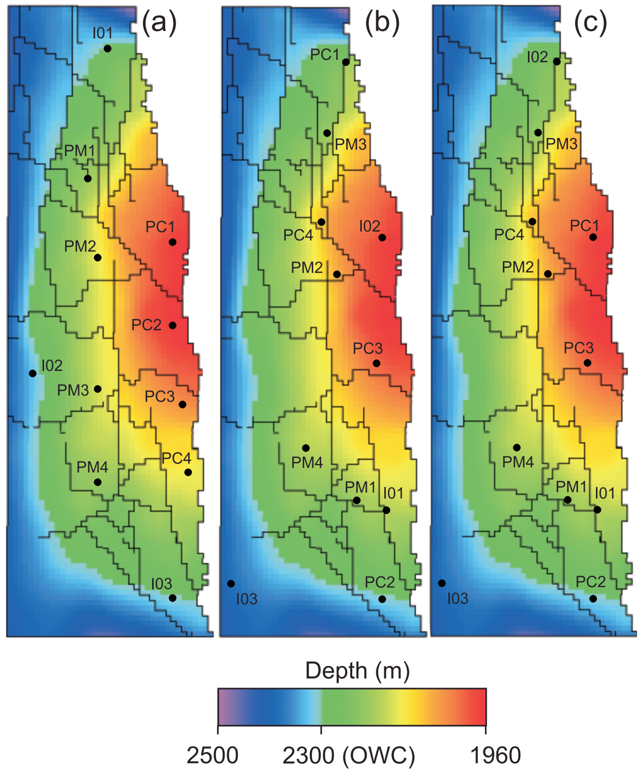

Fig. 6. Well positions for three development plans: (**a**) Plan-1, (**b**) Plan-2 and (**c**) Plan-3.

Reservoir engineering performance

Figure 6a shows the position of the wells in the original, hand-designed, development plan for the structure/fault pattern of the test reservoir (Plan-1). There is a line of crestal producers, another line of mid-structure producers and three edge injectors. Figure 6b shows the wells in the optimized development plan (Plan-2). Some wells have unusual positions: the injector I02 is at the crest of the reservoir, the producer PC2 has been placed in the water zone, the producer PC1 is only just in the oil leg, and the injector I01 is very close to the producer PM1. There are a number of important faults: one separates I01 from PM1, another separates I02 from the producers PM2, PC4 and PM3. Finally, Figure 6c shows the optimal plan but with I02 and PC1 swapped around, putting a producer back at the crest and an injector back at the oil–water contact (Plan-3).

The results of applying these three plans to the test reservoir are shown in Figure 7. The optimized development plan (Plan-2) produces 37% more oil than the base case (Plan-1). The revised plan (Plan-3) does noticeably less well than the optimized plan. It is concluded that the unusual positioning of an injector at the crest is quite important. This raises the question as to what is happening in this region of the reservoir. There are two particular elements of the optimized plan that need noting.

- The water from injector I02 is generally pushing oil along the geological layers down towards the producer PC1. A little water is crossing the fault that separates I02 from PM2, PC4 and PM3, and it provides pressure support to producers PM2 and PC4.

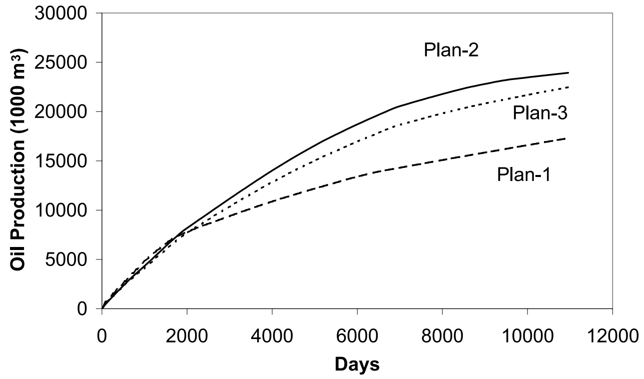

Fig. 7. Total oil production against time for the three tested development plans.

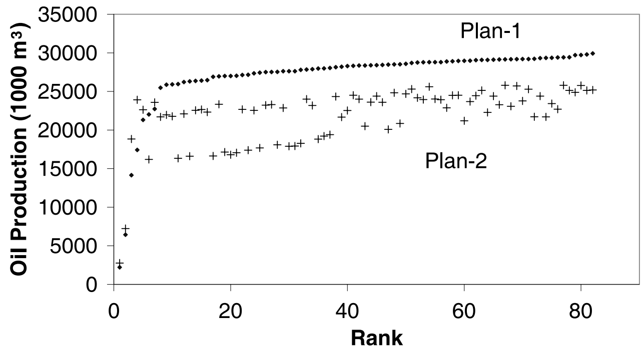

Fig. 8. A comparison of the total oil production obtained using Plan-1 and Plan-2 for the 82 reservoirs that had the same sedimentology as the test problem, but with different structural/fault patterns and fault properties.

- The water from I01 is driving oil towards producers PC3 and PM2, with little water crossing the fault towards producers PM1 and PM4.

From this analysis of the waterflood, one can see that the optimizer has managed to find a production plan that exploits particular features of the test reservoir.

An obvious question is whether Plan-2 has any value beyond this particular reservoir model. There are three possibilities to consider: Plan-2 may be exploiting some feature of the sedimentology which means that production from any of the reservoirs based on the same sedimentology could be improved; Plan-2 is exploiting some feature of the structure/fault pattern, again offering the possibility of improving the production across a range of reservoirs; finally, there is the possibility that Plan-2 works only for this reservoir and has no utility for other reservoirs.

Figure 8 compares the production, using Plan-1 and Plan-2, from the 82 reservoirs in the original sample that are based on the same sedimentology as in the test problem. The reservoirs cover a range of structures/fault patterns and fault properties, including an unfaulted version. The hand-designed development plan (Plan-1) was not necessarily planned to work with the structure used. The optimized development plan (Plan-2) was planned exclusively on the test reservoir. In most cases Plan-2 does significantly worse than Plan-1; only in six cases is there an improvement. It is concluded that Plan-2 is not exploiting a feature of the sedimentology.

Figure 9 compares the production, using Plan-1 and Plan-2, from the 243 reservoirs that use the same structure/fault pattern (nine different sedimentologies are used) but different fault properties (i.e. permeability and strain levels; see

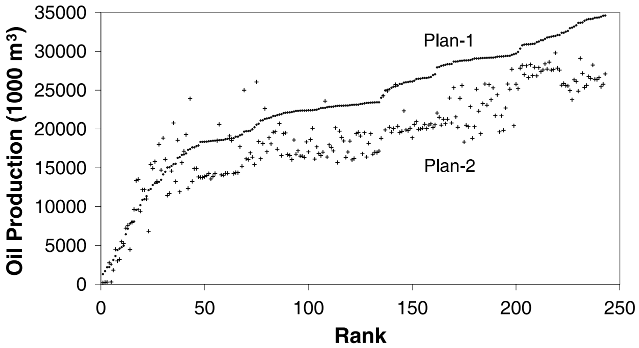

Fig. 9. A comparison of the total oil production obtained using Plan-1 and Plan-2 for the 243 reservoirs that had the same structural/fault patterns as the test problem, but with different sedimentology and fault properties.

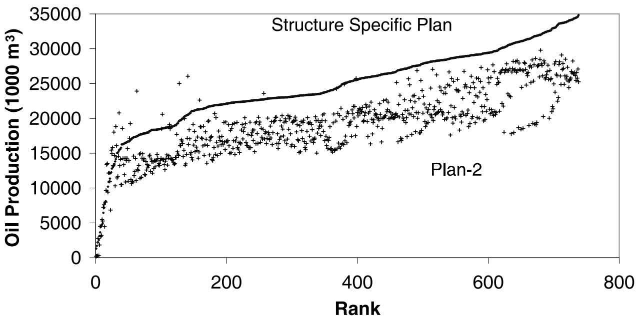

Fig. 10. A comparison of the total oil production obtained using either Plan-2 or the hand-tailored development plan that was tuned to the structural/fault patterns for each of the 738 sample reservoirs.

Manzocchi *et al.* 2008*a*). In the majority of cases Plan-2 produces a worse result. However, in 31 cases there is an improvement in the production, with most of these being in the bottom 20% of performers using Plan-1.

Finally, the result of using Plan-2 was compared against the appropriate hand-designed development plan on all 738 reservoirs. These reservoirs use nine different sedimentologies, four patterns of faults and many fault properties. The results are shown in Figure 10; the only reservoirs to show an improvement are the 31 that were identified in Figure 9. In all cases the faults are slightly leaky, with the production improved by up to 40% by using Plan-2.

It is observed that the optimized development plan works for a range of reservoirs with the same structure/fault pattern as in the test case and slightly leaky faults. If the faults are too open then water can flood through, this has the effect of killing producers PM2 and PC4 and less oil is produced by the end of the predefined production period (30 years). If the fault is totally sealing, then no water crosses the barrier and the pressure in the crestal region around the producers drops, which causes a reduction in well productivity. The effectiveness of the development plan appears not to depend on the sedimentology, with eight of the nine sedimentologies used appearing in the list of improved reservoirs. If an economic measure of the value of the optimized plan had been used, the situation may have looked even better. This is because the optimized plan uses one production well less that the original plan, which represents a significant financial saving.

In conclusion, the optimized plan is robust to uncertainties in the sedimentology and uncertainties in the fault properties, provided that the faults can be described as slightly leaky. The plan is not robust to significant changes in the pattern of faulting.

CONCLUSIONS AND DISCUSSION

This paper has shown that it is both possible, and feasible, to use a computer-based optimization method to plan the positions of all of the wells needed to produce economically from a reservoir, without the need to resort to using either reduced models or proxy models. It has been demonstrated that one can use a GA adapted to use a multi-processor system, and that it produces better results than a random search. The total number of solutions that were possible for the problem considered is approximately 1.3×10^{31}; the GA evaluated just 1650 solutions. The GA was run just once and no other optimization algorithm was tested, so it cannot be claimed that this is the best way of carrying out this type of optimization. However, the choice of the algorithm was guided by previous research (Ballester & Carter 2003).

The objective of the optimization was to maximize the total oil produced over a thirty-year period for a particular reservoir. This objective was chosen so as to be easy to extract from the simulation results. There is no reason to believe that the method would not work for an economics-based measure of success. The plan that was obtained was successful in that the production was increased by 37% over the thirty years. It had an unusual feature, with an injection well being placed at the crest of the reservoir. Analysis of the behaviour of the development plan, over a range of reservoirs, has shown that the plan was effective when the faults had particular locations and a particular range of fault properties. It appears that the effectiveness of the plan was largely independent of the sedimentology.

The European Commission is thanked for partly funding SAIGUP under the EU fifth Framework Hydrocarbons Reservoir Programme. A number of members of the earlier PUNQ project provided useful starter ideas on reservoir engineering for SAIGUP. We are grateful to Badley Geoscience, BG International, Roxar and Shell for their support of the SAIGUP project and for their sponsorship of the production of this thematic set.

REFERENCES

Aanonsen, S.I., Eide, A.L., Holden, L. & Aasen, J.O. 1995. Optimizing reservoir performance under uncertainty with application to well location. Paper SPE 30710, presented at the 70th Annual SPE Technical Conference and Exhibition, Dallas, October.

Ballester, P.J. & Carter, J.N. 2003. Real-parameter genetic algorithms for finding multiple optimal solutions in multi-modal optimization. *Lecture Notes in Computer Science,* **2723**, 706–717.

Beckner, B.L. & Song, X. 1995. Field development planning using simulated annealing: Optimal economic well scheduling and placement. Paper SPE 30650, presented at the 70th Annual SPE Technical Conference and Exhibition, Dallas, October.

Bittencourt, A.C. & Horne, R.N. 1997. Reservoir development and design optimization. Paper SPE 38895, presented at the 72nd Annual SPE Technical Conference and Exhibition, San Antonio, October.

Carter, J.N. 2003. Introduction to using genetic algorithms. *In:* Nikravesh, M., Aminzadeh, F. & Zadeh, L.A. (eds) *Soft Computing and Intelligent Data Analysis in Oil Exploration.* Developments in Petroleum Science, **51**. Elsevier Science, Oxford, 51–76.

Centilmen, A., Ertekin, T. & Grader, A.S. 1999. Applications of neural networks in multiwell field development. Paper SPE 56433, presented at the 74th Annual SPE Technical Conference and Exhibition, Houston, October.

Craig, F.F. 1971. *The Reservoir Engineering Aspects of Water-flooding.* SPE Monograph, **30**.

Deb, K. 2001. *Multi-objective Optimization Using Evolutionary Algorithms.* John Wiley and Sons, Chichester.

Deb, K. & Agrawal, R.B. 1995. Simulated binary crossover for continuous search space. *Complex Systems,* **9**, 115–148.

Deb, K. & Beyer, H.-G. 1999. Self adaption in real parameter genetic algorithms with simulated binary crossover. *In:* Banzhaf, W., Daida, J., Eiben, A.E., Garzon, M.H., Honavar, V., Jakiela, M. & Smith, R.E. (eds) *Proceedings of the Genetic and Evolutionary Computation Conference.* Morgan Kauffman, San Francisco, 172–179.

Goldberg, D.E. & Deb, K. 1991. A comparative analysis of selection schemes used in genetic algorithms. *In:* Rawlins, G. (ed.) *Foundations of Genetic Algorithms.* Morgan Kaufmann, San Francisco, 69–93.

Guyaguler, B. & Gumrah, F. 1999. Comparison of genetic algorithm with linear programming for the optimization of an underground gas-storage field. *In Situ,* **23**, 131.

Guyaguler, B., Horne, R.N., Rogers, L.L. & Rosenweig, J.J. 2002. Optimization of well placement in a Gulf of Mexico water-flooding project. *SPE Reservoir Evaluation & Engineering,* **June**, 229 (Paper SPE 78266).

Howell, J.A., Skorstad, A., MacDonald, A., Fordham, A., Flint, S., Fjellvoll, B. & Manzocchi, T. 2008. Sedimentological parameterization of shallow-marine reservoirs. *Petroleum Geoscience,* **14**, 17–34.

Manzocchi, T., Carter, J.N., Skorstad, A. *et al.* 2008*a*. Sensitivity of the impact of geological uncertainty on production from faulted and unfaulted shallow-marine oil reservoirs: objectives and methods. *Petroleum Geoscience,* **14**, 3–15.

Manzocchi, T., Matthews, J.D., Strand, J.A. *et al.* 2008*b*. A study of the structural controls on oil recovery from shallow-marine reservoirs. *Petroleum Geoscience,* **14**, 55–70.

Matthews, J.D., Carter, J.N., Stephen, K.D., Zimmerman, R.W., Skorstad, A., Manzocchi, T. & Howell, J. 2008. Assessing the effect of geological uncertainty on recovery estimates in shallow-marine reservoirs: the application of reservoir engineering to the SAIGUP project. *Petroleum Geoscience,* **14**, 35–44.

Mitchell, M. 1998. *An Introduction to Genetic Algorithms.* MIT Press, Cambridge.

Pan, Y. & Horne, R.N. 1998. Improved methods for multivariate optimization of field development scheduling and well placement design. Paper SPE 49055, presented at the 73rd Annual SPE Technical Conference and Exhibition, New Orleans, September.

Rogers, L.L. & Dowla, F.U. 1994. Optimal ground-water remediation using artificial neural networks with parallel solute transport. *Water Resources Research,* **30**, 458.

Thakur, G.C. 1996. What is Reservoir Management? *Journal Petroleum Technology,* **48**, 520–525.

Young, L.C. & Hemanth-Kumar, K. 1991. High Performance Black Oil Calculations. *Proceedings of the 11th SPE Symposium on Reservoir Simulation*, Paper SPE 21215.

Received 29 January 2007; revised typescript accepted 29 September 2007.