IOP Conf. Series: Journal of Physics: Conf. Series **1268** (2019) 012029 doi:10.1088/1742-6596/1268/1/012029

Comparison of design optimization algorithms of a multiply fractured horizontal well

E A Kavunnikova[1,2], B N Starovoitova[1], S V Golovin[1,2] and A M Krivtsov[3]

[1] Lavrentyev Institute of Hydrodynamics, Lavrentyev pr. 15, Novosibirsk, Russia
[2] Novosibirsk State University, Pirogova str. 1, Novosibirsk, Russia
[3] Peter the Great St.Petersburg Polytechnic University, Polytechnicheskaya str. 29, St.Petersburg, Russia

E-mail: ekavunnikova@gmail.com, botagoz@hydro.nsc.ru, golovin@hydro.nsc.ru

Abstract. The paper is devoted to comparison of multiple-objectives optimization algorithms in application to the problem of design optimization of a multiply fractured horizontal well (MFHW). The problem is stated either as a single-objective one, where only the income based on Net Present Value (NPV) is maximized, or as a multi-objective problem, where it is necessary to simultaneously find extremes of NPV, the post-fracture oil production and fracturing costs. Three popular stochastic optimization methods are considered: genetic algorithms (GA), simulated annealing (SA) and particle swarm optimization (PSO). Since PSO, SA and GA techniques employ different strategies and computational efforts, the comparison of their efficiency was carried out by testing on synthetic problems and then applied to the example of a MFHW in a low-permeable oil reservoir.

1. Problem definition

Multiply fractured horizontal wells (MFHW) are widely used for enhancing the oil recovery of low permeable reservoirs. Because of the high cost of MFHW, it is of great interest to study the optimization problem of hydraulic fracturing design to ensure its economic efficiency. At that, the choice of a fast and stable optimization algorithm plays a crucial role.

The optimization problem of hydraulic fracturing design is formulated as follows. One should choose the length of a horizontal well, the number of fractures and the geometric characteristics of each fracture, at which the following optimization targets are achieved:

- the maximum of cumulative well production : $Q_{tot} \to \max$;
- the maximum of NPV-based income : $NPV \to \max$;
- the minimum treatment costs of MFHW: $C_{HF} \to \min$.

Both, the single-objective optimization problem, where only one function (NPV) is maximized, and the multi-objective optimization problems, where it is necessary to simultaneously find extremes of three functions (NPV, C_{HF} and Q_{tot}) are considered.

Mathematically, any optimization problem can be stated as follows: find $\mathbf{x} = (x_1, \dots, x_n)$ to maximize or minimize objective functions $(f_1(\mathbf{x}), \dots f_M(\mathbf{x}))$, subject to bounds $x_{L,i} \leqslant x_i \leqslant x_{U,i}$. Here $\mathbf{x}$ represents the vector of free optimization parameters; $f_1(\mathbf{x}), \dots, f_M(\mathbf{x})$ are objective functions or criteria; $x_{L,i}$, $x_{U,i}$ are lower and upper bounds correspondingly. The set of all

possible point $\mathbf{x}_i$, which are potential solutions of the optimization problem, is called the search space. Since $\max(f_i) = -\min(-f_i)$, we consider only the minimization problem.

If the number of criteria M is equal to one, then we have a single-objective optimization problem (SOOP) or the problem of finding the global mimimum value of $f(\mathbf{x})$. In this case, it is easy to compare potential solutions: if $f(\mathbf{x}_1) < f(\mathbf{x}_2)$, then the potential solution $\mathbf{x}_1$ is better than $\mathbf{x}_2$. If $M > 1$, then we have a multi-objective optimization problem (MOOP). In this case the concept of domination is generally used to compare two solutions $\mathbf{x}_1$ and $\mathbf{x}_2$. It is said that $\mathbf{x}_1$ dominates $\mathbf{x}_2$, if $\mathbf{x}_1$ is not worse than $\mathbf{x}_2$ for all objective functions ($f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2), i = 1, \ldots, M$) and at least for one objective is better than $\mathbf{x}_2$ ($\exists k \in [1, \ldots, M] : f_k(\mathbf{x}_1) < f_k(\mathbf{x}_2)$). On the other hand, if $\mathbf{x}_1$ is better than $\mathbf{x}_2$ for one criterion, but $\mathbf{x}_2$ is better than $\mathbf{x}_1$ for another one, then it is said that $\mathbf{x}_1$ and $\mathbf{x}_2$ are non-dominant. The set of non-dominate solutions is called Pareto front.

For MOOP it is often impossible to find a unique $\mathbf{x}$ that provides minimum values of all objective functions simultaneously, because the criteria may contradict each other. The solution of MOOP is the set of optimal solutions each one being better than another one for at least one criterion, and there exist no other solutions in the entire search space, which dominate any member of this set. The set of optimal solutions of MOOP is called the optimal Pareto front.

We consider the most popular and reliable optimization algorithms: genetic algorithm NSGA-II, the particle swarm optimization method and the simulated annealing method.

2. Optimization algorithms

2.1. Genetic algorithms.

GA presented by John Holland in 1975, is an evolutionary search algorithm that emulates the process of natural selection. For GA, a set of potential solutions is called a population and each member of the population is called an individual. Any of GA starts from a randomly generated initial population. The changes of the population from one generation to another occur under the influence of operators of crossover, mutation and selection. The operator of selection emulates the principal of "survival of the fittest". The operator of crossover simulates the reproduction processes. The operator of mutation describes small changes in characteristics of an individual and provides the diversity in the population. After applying these operators, the new population is created. The process is iterated until the stopping criterion is met, for instance, until the number of generations (iterations) reaches a given value. There are many references concerning the details of the GA (e.g., [1]). In this article, one of the popular algorithms NSGA-II (Non-dominated Sorting Genetic Algorithm-II) proposed in [2] is applied.

The feature of the NSGA-II is the selection operator for the best individuals. In NSGA-II, the current population is sorted on the non-domination criterion and each individual $\mathbf{x}_i$ is assigned a rank according to the rule: $\text{rank}(\mathbf{x}_i) = 1 + nD_i$, where nD_i is a number of individuals dominating $\mathbf{x}_i$. So, the non-dominant set of individuals in the current population has the rank equal to one and forms the first front. The individuals, which have the rank equals 2, are dominated only by the individuals of the first front and form the second front and so on.

In addition to the rank, a crowding distance for individuals of the same rank [2] is defined. The crowding distance for the i-th individual is equal to the normalized perimeter of a cuboid with two diagonal vertices taken as $i-1$-th and $i+1$-th individuals. The greater the crowding distance is, the better diversity in the population is obtained. It is said that the individual $\mathbf{x}_1$ is better than the individual $\mathbf{x}_2$ if either $\text{rank}(\mathbf{x}_1) < \text{rank}(\mathbf{x}_2)$, or $\text{rank}(\mathbf{x}_1) = \text{rank}(\mathbf{x}_2)$ and the crowding distance of $\mathbf{x}_1$ is larger than the crowding distance of $\mathbf{x}_2$.

Notice, that for SOOP the crowding distance is determined in the parameter space, whereas for MOOP the distance is calculated in the criteria space.

2.2. Particle swarm optimization

The PSO method proposed in [3] is based on the simulation of the behaviour of birds within a flock. The swarm consists of a set of particles, each one representing a potential solution. The i-th particle ($i = 1, ..., N$) is determined by its position $\mathbf{x}_i$ in the parameter space, the velocity $\mathbf{v}_i$ and the state function $f(\mathbf{x}_i)$. PSO, as well as GA, starts with generating an initial swarm. The particle position changes according to its own experience and the experience of the entire swarm:

$$\begin{aligned} \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{v}_i(t+1), \\ \mathbf{v}_i(t+1) &= C_{in}\mathbf{v}_i(t) + C_{cog}r_1[\mathbf{x}_{pbest,i}(t) - \mathbf{x}_i(t)] + C_{soc}r_2[\mathbf{x}_{gbest}(t) - \mathbf{x}_i(t)]. \end{aligned} \tag{1}$$

Here t indicates a pseudo-time that is the iteration step; $\mathbf{x}(t)$ is the particle's position at the time t; $\mathbf{v}(t)$ is the velocity of the i-th particle at the time t; $\mathbf{x}_{pbest,i}$ is the personal best position for the i-th particle at the time t; $\mathbf{x}_{gbest}$ is the best position for the whole swarm at the time t; C_{in} is the inertia factor, which shows the influence of the previous direction; C_{cog} is a cognitive coefficient reflecting the aspiration of the particle to move towards its best position; C_{soc} is a social coefficient that corresponds to the movement towards the most successful particle in the swarm; r_1, r_2 are random numbers uniformly distributed on $(0, 1)$.

In this article, coefficients C_{in}, C_{cog} and C_{soc} are considered to be random values in the following intervals: $C_{in} \in (0.1, 0.5)$, $C_{cog} \in (1.5, 2.0)$ and $C_{soc} \in (1.5, 2.0)$. Furthermore, the particle may be affected by turbulence (the full analogue of the mutation operator in GA). We use the mutation scheme of [4]: 1/3 of the swarm undergoes a uniform mutation, and 1/3 — a non-uniform mutation, whereas the rest does not mutate.

In order to apply PSO to multi-objective cases, PSO must be modified. The modifications to PSO algorithm concern the selection of the personal best (*pbest*) and the global best (*gbest*) for a particle. In this paper, the approach based on Pareto dominance and a crowding factor [4, 5] is used, and is referred to as OMOPSO. In OMOPSO, the set of current non-dominated solutions is considered as the best solutions of the entire swarm and is stored separately in an external archive called leaders. The maximum size of the leaders may be less than or equal to the swarm size. At each iteration, for each particle one chooses $\mathbf{x}_{gbest}$ from leaders, finds a new position by (1), and evaluates the new objective functions. The new position of a particle replaces its current *pbest* if the new position dominates the previous one, or if both are non-dominated with respect to each other. After the update of all the particles, the leaders are also updated by the particles that have improved their *pbest* value. If the leaders' size becomes larger than required, its size is reduced by the selection procedure using the non-dominated sorting and the crowding distance calculation as in NSGA-II. The rest of the archive is eliminated. The feature of this algorithm is the simplicity of control over the leaders' size, which does not require any additional criteria.

2.3. Simulated annealing

Single-objective optimization by SA is based on the thermodynamic analogy with the metal cooling and annealing processes. If the cooling is slow enough, the energy state of metals at the end will be close to its minimum value. The search space describes a set of states of the physical system, and $f(\mathbf{x})$ corresponds to the system energy. SA uses a point by point search technique opposed to the population-based search of GA or PSO. It starts with a randomly generated point in the search space. At each iteration, the point undergoes some random perturbation to produce a new solution. If the energy decreases, the new solution is accepted as the state and the search continues from this point. However, if the energy increases, the new solution is accepted with the acceptance probability p given by: $p(\Delta f, T) = 1/(1 + \exp(\Delta f/T))$ or $p(\Delta f, T) = \exp(-\Delta f/T)$. Here Δf is the system energy difference; T is a control parameter that, by analogy, is referred to as the temperature of the system.

The algorithm proceeds a certain number of iterations at each temperature (inner loop) and then the temperature decreases in accordance with the cooling schedule. Notice that the feature of SA is that the search can potentially move away from the point of local minimum. For high temperatures, a new solution is most likely to be accepted as a state irrespective of the sign $-\Delta f$. As the temperature drops, the probability for accepting a new solution becomes small. So, the cooling schedule which defines the acceptance probability must be sufficiently slow. There are different cooling schedules among which are Boltzmann (logarithmic), Cauchy and quenching (exponential) (for more details see [6]). In this article, the popular geometric cooling schedule given by $T_k = T_0 \cdot \alpha^k$,, where k is the iteration step, T_0 is the initial temperature, $\alpha \in [0.5, 0.99]$ is the cooling factor, is used. Iterations continue until the temperature reaches the value T_{end}.

The key issues in extending SA for MOOP are to determine the system energy and the probability of acceptance. In the article, it is used the algorithm AMOSA proposed in [7] as a generalization of SA to the multi-objective case with the exception of the selection procedure, that is chosen as in NSGA-II. In AMOSA, similar to the multi-objective PSO, the best solutions are stored in an external archive, which has two limits: a soft limit (SL) and a hard limit (HL). At first, a set of potential solutions is randomly generated, and all the non-dominate solutions are chosen. If the number of non-dominate solutions exceeds HL, they are reduced to HL and are stored in the archive. A randomly selected solution from the archive is then taken as the starting point for the optimization. The system energy is approximated by the dominance measure. For two given states a and b the energy difference is defined as $\Delta E_{a,b} = \prod_{i=1, f_i(a) \neq f_i(b)}^{M} (|f_i(a) - f_i(b)|)/R_i$, where M is the number of objectives and R_i is the range of the i-th objective. The acceptance probability depends on the domination status of the new solution with respect to the current state and all points in the archive. During the optimization process, the new non-dominate solutions are added to the archive until its size exceeds SL. Then, the archive size is reduced to HL. At the end of iterations, the archive represents the solution.

3. Application to the test problems

Before applying the considered algorithms for the MFHW optimization, they were tested on synthetic problems. It should be noted that all the parameters used in the methods were determined after the set of calculations, which are omitted here. In the article, the real-coded NSGA-II algorithm consisting in the direct representation of the actual values of optimization parameters is used. For OMOPSO, the size of leaders is equal to the size of the whole swarm. For multi-objective and single–objective SA, the cooling schedule is taken as $T_k = T_0 \cdot \alpha^k$ with $\alpha = 0.95$ and T_0 chosen so that the initial acceptance probability is greater than 50%. In numerical experiments, 20 independent runs are performed and metrics for evaluating the performance are calculated.

3.1. Test function for single-objective optimization

Rastrigin function is a strongly multimodal function with regularly distributed local minima:

$$F_1(\mathbf{x}) = 10n + \sum_{i=1}^{n} [x_i^2 - 10 \cdot \cos(2\pi x_i)], \quad n = 3, \quad -5.12 \leq x_i \leq 5.12.$$

Global minimum $f(0, 0, 0) = 0$. The cross-section of the Rastrigin function is shown in figure 1.

All the considered optimization methods are run for a maximum of 10000 function evaluations. To do this, the size of population for NSGA-II and the swarm size for PSO are set to 100 and the iteration number (the generation number in NSGA-II) is chosen as 100. For NSGA-II, the whole population takes part in the reproduction process; the arithmetic crossover and the mutation of a Gaussian type with $\sigma = 0.5$ are applied. Two variants of PSO with Gaussian mutation (PSOm) and without mutation (PSO) are considered. For SA, the number of temperature steps and the size of the inner loop are equal to 100; the new solution is generated either in

accordance to the Gaussian distribution with $\sigma = \sqrt{T}$ (represented by $\mathrm{SA_B}$) or in accordance to the Cauchy distribution (represented by $\mathrm{SA_C}$). Since SA depends on the starting point, a case with a refinement of the starting solution, for example, by the simple hill climbing technique represented by $\mathrm{SA_{CR}}$, is considered.

Table 1 gives the performance results: the mean computation time $\bar{t}$ and the mean value of $\overline{F_1}$. Notice that the result obtained with NSGA-II is better than those obtained with others but it is slower. Figure 2 shows the dependence of the mean value of F_1 on the number of objective evaluations. One can see that NSGA-II converges faster than another methods.

	NSGA-II	PSOm	PSO	$\mathrm{SA_B}$	$\mathrm{SA_C}$	$\mathrm{SA_{BR}}$
$\overline{F_1}$	0.002246	0.269221	0.795967	0.7086	0.075076	0.003017
$\overline{t_1}$	69.330165	0.475257	0.212312	0.261867	0.171905	0.27312

Table 1. Minimization of Rastrigin function.

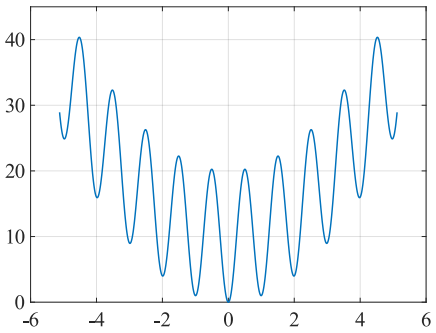

Figure 1. The cross-section of Rastrigin function.

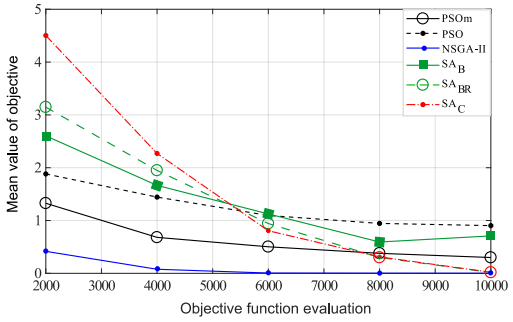

Figure 2. Dependence of the mean value of the objective function on the number of evaluations.

3.2. Test calculations for multi-objective optimization

For MOOP, the algorithms are tested on a complicated DTLZ problems, namely, the DTLZ4 one [8] for three objective functions:

$$
\begin{aligned}
&(F_1, F_2, F_3) \to \min, \quad g(\mathbf{x}) = \textstyle\sum_{i=3}^{N}(x_i - 0.5)^2,\ N = 12,\ 0 \le x_i \le 1,\\
&F_1(\mathbf{x}) = (1+g)\cos(0.5\pi x_1^{100})\cos(0.5\pi x_2^{100}),\\
&F_2(\mathbf{x}) = (1+g)\cos(0.5\pi x_1^{100})\sin(0.5\pi x_2^{100}), \quad F_3(\mathbf{x}) = (1+g)\sin(0.5\pi x_1^{100}).
\end{aligned}
\tag{2}
$$

Optimal Pareto front of (2) is the part of the sphere $f_1^2 + f_2^2 + f_3^2 = 1$, where $f_i \ge 0$.

For all algorithms, the value of each function evaluations is taken as 50000. The population size and the number of generation in NSGA-II are taken to be equal to 200 and 250, respectively. The swarm size and the leaders size are chosen as 200 particles, and the iterations number equals 250 is used in OMOPSO. For AMOSO, the inner loop was increased to 500 iterations, the archive sizes HL and SL are set to 200 and 300, respectively. Moreover, for AMOSA the simple hill climbing technique to refine the initial solutions in the archive is used.

For evaluating the performance of the methods, two metrics are calculated: Generational Distance (GD) and Spacing (S). The GD criterion measures the average distance between the points of the obtained Pareto front (PF) and the true optimal Pareto front (PF^*): $GD = \left(\sum_{i=1}^{n_{PF}} d_i^2\right)^{1/2}/n_{PF}$. Here n_{PF} is the number of PF points ($n_{PF} = |PF|$); d_i is the Euclidean distance (in the space of criteria) between the i-th member of PF and the nearest member of PF^*. In the ideal case, $GD = 0$. The criterion S characterizes the distribution of solutions along the PF and is defined by the standard deviation of the absolute differences between the i-th and its nearest solutions of PF: $S = \sqrt{\left(\sum_{i=1}^{n_{PF}} (D_i - \overline{D_i})^2\right)/n_{PF}}$. Here $D_i = \min_{j,j\neq i}(|f_1^i - f_1^j| + \ldots + |f_M^i - f_M^j|)$, and $\overline{D}$ is the mean value of D_i. Note, that D_i is different from Euclidean distance between two solutions. The smaller S, the better is the algorithm.

For NSGA-II, OMOPSO and AMOSA, the average GD is equal to 0.022, 0.02776 and 0.014526, respectively; the average S is equal to 0.1493, 0.17145 and 0.118, respectively. Figures 3–5 show the true Pareto fronts and the final approximations of PF^* obtained by NSGA-II, OMOPSO and AMOSA, correspondingly. Notice that the values of GD and S are quite acceptable for all three methods. However, OMOPSO could not find the Pareto front completely and AMOSA just obtained the boundary points of PF (see fig. 3–5). In this case, the good performance is explained by the fact that the points obtained by AMOSA are very close to the true Pareto front.

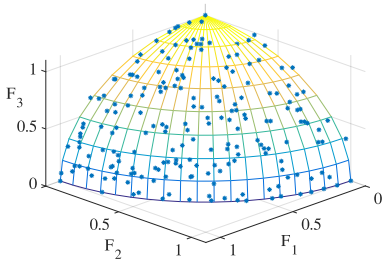

Figure 3. PF by NSGA-II.

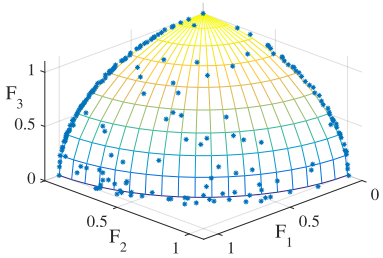

Figure 4. PF by OMOPSO.

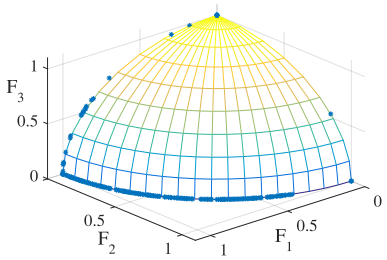

Figure 5. PF by AMOSA.

4. Application to the optimization of MFHW design

4.1. Optimization parameters and objective functions

Since the geometric characteristics of the fracture depend on the amount of proppant injected into the reservoir, we choose the horizontal well length, the number of fractures and the amount of proppant as optimization parameters under the following bound constraints:

$$4 \leq N_f \leq 12, \quad 4000 \leq M_p \leq 90000 \text{ [kg]}, \quad 400 \leq L_w \leq 1200 \text{ [m]}, \tag{3}$$

where N_f is the number of fractures; M_p is the proppant mass required to create one fracture; L_w is the length of the horizontal well.

To evaluate the objective functions, it is necessary to solve three interrelated problems: the dependence of the fracture geometry on the amount of injected proppant, the calculation of the post-fracture oil production and the calculation of economic parameters. In this article, an approach proposed in [9] is used. Note that all the assumptions of [9] are valid in this case.

To calculate the fracture geometric characteristics, a fast algorithm for a penny-shaped hydraulic fracture under the standard linear theory assumptions on reservoir properties [10] is applied.

According to [11] and [9], the production of the horizontal well with multiple fractures is approximated by the formula $Q_{tot} = Q(1 - e^{-\alpha t})/\alpha$, where α is an annually decline rate, t is the time for which the well production is calculated, Q is the annual production rate.

NPV is determined by the equality $NPV = \sum_{t=1}^{T_{\max}} (\Pi_t - A_t)/(1 + D)^t - C_{HF}$. Here $T_{\max}$ is the time period for the revenue calculation; t is the current year; Π_t is the revenue from the oil sale at t-th year; A_t is current expenses, which include operational costs, taxes, costs of oil treatment and transportation; D is the discount rate; C_{HF} is the fracturing costs, which include the costs of proppant and fracturing fluid, their injection costs and some additional costs (for example, the cost of equipment).

4.2. Results of MFHW optimization

If we consider only the problem of maximizing NPV, we get the single-objective optimization problem with constraints (3). In this case, we use the algorithms parameters as for Rastrigin function and only one variant of SA : $\mathrm{SA_B}$ with the refinement of a starting point. All the considered methods give similar results and the average values are shown in Table 2. Notice that the average computation times are comparable.

	N_f	M_p, kg	L_w, m	NPV, \$	t
NSGA-II	9	90000	877	$9.79 \cdot 10^5$	248.66
PSO	9	90000	877	$9.79 \cdot 10^5$	239.33
PSOm	9	90000	877	$9.79 \cdot 10^5$	256.54
$\mathrm{SA_B}$	9	89325	872	$9.77 \cdot 10^5$	286.15

Table 2. Results of maximisation of NPV under constrains (3).

If we search for extrema of three objective functions NPV, Q_{tot} and C_{HF} simultaneously, we get the multi-objective optimization problem with constraints (3). The number of each function evaluations is set to 20000. The population size for NSGA-II, both the swarm and the leaders sizes in OMOPSO are set to 200 and the iteration number is taken as 100. For AMOSA, the inner loop is 200 iterations, the HL and SL values are equal to 100 and 200, respectively, and the refinement of the initial archive is used. The number of temperature steps and the end temperature are chosen so that the number of each function calculations does not exceed 20000. In this case, the GD metric cannot be used because of the true Pareto front is unknown. Instead, the Dimensional Extent (DE) metric according to [12]: $DE = \sqrt{\sum_{m=1}^{M} \max |f_m(a) - f_m(b)|}$, $a, b \in PF$ is adopted. DE estimates the distance between the most distant solutions of PF so that a wider spread produces a larger DE.

The average values of S, DE and the computation times are represented in Table 3. The calculations demonstrate that all algorithms give comparable results with respect to S metric. As for DE, AMOSA gives the worst result, i.e. the spread of PF obtained by the AMOSA method is smaller than by two others. This is evident from figures 6-8, which present the final PF obtained by OMOPSO, AMOSA and NSGA-II, respectively. Although the computation time is the best for OMOPSO, NSGA-II converges faster then other two methods. Therefore, in practice, there is no need to calculate all 100 generations. AMOSA is the slowest algorithm due to the need to sort the large archive and the initial refinement procedure.

It should be noted, that NPV optimization gives only one point whereas the MOOP gives the whole picture of profit-investment relationships. The final decision should be made by a person on the base of additional preferences.

NSGA-II			OMOPSO			AMOSA		
S	DE	t	S	DE	t	S	DE	t
0.0189	2.6199	350.52	0.0167	2.6199	340.54	0.0161	2.5618	503.87

Table 3. Performance metrics for NSGA-II, OMOPSO, and AMOSA.

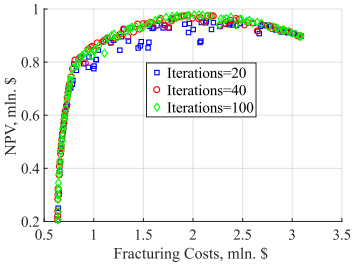

Figure 6. PF by OMOPSO.

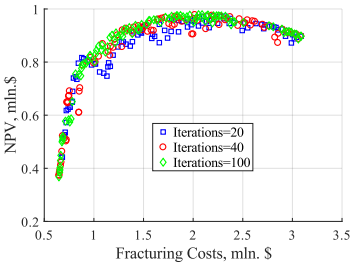

Figure 7. PF by AMOSA.

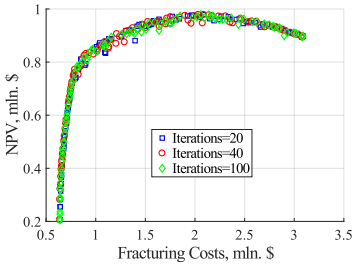

Figure 8. PF by NSGA-II.

5. Conclusion

Three popular optimization methods (NSGA-II, PSO, SA) are considered. Test problems revealed a strong dependence of SA convergence on its parameters. However, in the real optimization problem the algorithm requires large number of objective function calculations. For the optimization problem of hydraulic fracturing design, it seems that two optimization methods PSO and NSGA-II are of interest. NSGA-II showed the ability to determine complex Pareto front, whereas PSO obtains acceptable computation time.

Acknowledgments

This work was supported by Ministry of Education and Science of the Russian Federation, grant No. 14.575.21.0146, unique identifier: RFMEFI57517X0146. Authors are grateful to the Gazpromneft Science & Technology Centre for providing data used in numerical experiments.

References

[1] Deb K 2001 *Multi-objective optimization using evolutionary algorithms* vol 16 (Chichester, England: John Wiley & Sons)
[2] Deb K, Agrawal S, Pratap A and Meyarivan T 2002 *IEEE Trans. Evol. Comput.* **6** 182–197
[3] Kennedy J and Eberhart R 1995 *Proc. ICNN'95 — Int. Conf. on Neural Networks (27 Nov.-1 Dec. 1995, Perth, Australia)* vol 4 pp 1942–48
[4] Sierra M R and Coello C A C 2005 *Evolutionary Multi-Criterion Optimization. EMO 2005* vol 3410 ed Coello C A C, Hernandez A A *et al.* (Berlin, Heidelberg: Springer) pp 505–519
[5] Sierra M R and Coello C A C 2006 *Int. J. Comput. Intell. Res.* **2** 287–308
[6] Suman B and Kumar P 2006 *J. Oper. Res. Soc.* **57** 1143–1160
[7] Bandyopadhyay S, Saha S, Maulik U and Deb K 2001 *IEEE Trans. Evol. Comput.* **12** 269–283
[8] Deb K, Thiele L, Laumanns M and Zitzler E 2005 *Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing* ed Abraham A *et al.* (London: Springer) pp 105–145
[9] Starovoitova B N, Golovin S V, Paderin G V, Shel E V, Kavunnikova E A and Krivtsov A M 2018 *IOP Conf. Ser.: Earth Environ.Sci.* **193** 012011
[10] Dontsov E V 2016 *R. Soc.Open Sci.* **3** 160737
[11] Elkin S V, Aleroev A A, Veremko N A and Chertenkov M V 2016 *Neft. Khoz. – Oil Ind. J.(in Russ.)* **12** 110–113
[12] Zitzler E 1999 *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications* Ph.D. thesis Swiss Federal Institute of Technology Zurich