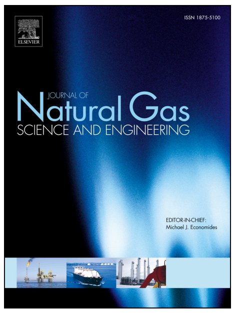

# Sperm Whale Algorithm: an Effective Metaheuristic Algorithm for Production Optimization Problems

A. Ebrahimi, E. Khamehchi

PII: S1875-5100(16)30001-4

DOI: 10.1016/j.jngse.2016.01.001

Reference: JNGSE 1205

To appear in: *Journal of Natural Gas Science and Engineering*

Received Date: 22 August 2015

Revised Date: 1 January 2016

Accepted Date: 5 January 2016

# Sperm Whale Algorithm: an Effective Metaheuristic Algorithm for Production Optimization Problems

**Ebrahimi A.[1], Khamehchi E.[1,*]**

[1]Faculty of Petroleum Engineering, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

## Abstract

A new optimization algorithm called **Sperm Whale Algorithm** (SWA) is proposed to solve production optimization problems. This algorithm is based on the sperm whale's lifestyle. Like other population-based algorithms, SWA uses a population of solutions to find the optimum answer. One of the advantages of this method over others is that it uses two contradictory types of answers: it uses the worst and the best answers to reach the optimum point. The SWA algorithm was tested on 26 benchmarks and three benchmarks in several dimensions and one production optimization problem. The results and comparison of its performance with other algorithms show that SWA's performance is superior to other algorithms and it could be confidently used in optimization tasks.

**Keywords:**

Optimization algorithms, Sperm Whale Algorithm, Optimization, Production Optimization

## 1. Introduction

Production optimization has been offered as a response to the increasing world demand for oil and gas. Optimization algorithms have many applications, and optimization techniques have proved very effective in various problems such as reserves development, well testing, distribution of natural gas reserves, design of conditions in operational production, development of production and injection projects, solving problems related to gas lift systems, and other problems. Simply stated, optimization is the selection of the best choice from among available options. Optimization algorithms can be divided into deterministic and stochastic classes. Deterministic methods themselves are classified into evaluation

[*] Corresponding author, E-mail: khamehchi@aut.ac.ir ,Tel: +982164543528(EX.4)

and gradient-based methods. In purely computational methods, there is no need for calculating gradient functions, but they are extremely slow and ineffective methods. Gradient-based algorithms use gradients or derivations of objective functions to direct the search. However, these methods do not guarantee convergence to the global optimum point, unless when the objective function is flat and smooth. Briefly, we can say that in deterministic methods, whether they use information on derivatives or receive help from methods without derivatives, the fact remains that the answers will still be local, the final answer will be greatly dependent on the initial guess, and they will fail if there are many local optimum points in the problem. In fact, the problems which have high dimensionality, multimodality, epistasis (parameter interaction), and non-differentiability are difficult or impossible to solve using this class of methods [1].

However, stochastic methods try to find the global optimum point. They do not require gradient information but rather search for the global optimum point through calling the objective function many times. These methods can be divided into several general classes: the methods of random search, evolutionary methods, intelligent population methods, and other methods such as harmony search, etc. [2]. However, in general, they can be divided into two categories: the single-based solution methods and the population-based solution methods [1, 3]. The following algorithms are classified in the single-based category:

Simulated Annealing (SA) [4], Tabu Search (TS) [5], Iterated Local Search (ITS) [6], Guided Local Search (GLS) [7], Pattern Search (PS) [8],Random Search (RS) [9], Variable Neighborhood Search (VNS) [10] and Vortex Search algorithm (VS)[11].

There are various population-base algorithms: the genetic algorithm (GA) based on Darwin's theory [12], the Particle Swarm Optimization (PSO) algorithm based on the swarm movement of birds [13], Differential Evolution (DE) [1, 14], Estimation of Distribution Algorithms (EDA) [15], Ant Colony Optimization (ACO) [16], Artificial Bee Colony algorithm (ABC) [17-20], Harmony Search [21], Bacterial Foraging Optimization Algorithm (BFOA) [22], League Championship Algorithm (LCA) [23], Firefly Algorithm (FA) [24, 25], Group Search Optimizer (GSO) [26], Cuckoo Search algorithm (CS) [27], Krill Herd algorithm (KH) [28], Artificial Chemical Reaction Optimization Algorithm (ACROA)

[29], Stochastic Fractal Search (SFS) [30], Symbiotic Organisms Search (SOS) [31] and Optics Inspired Optimization (OIO) [32] that are characterized by their names.

These algorithms find the solution by effectively searching the search space and reducing its size. The main difference between different algorithms is in fact in their approach to balance between exploration (global search capability) and exploitation (local search capability around the near-optimal solution) [3]. Most of the algorithms use the best sections of solution to reach an optimal solution and discard bad ones. In fact, they do not make utmost use of the existing data. That is why it seems necessary to develop an algorithm that uses all solutions, extracts maximum information from existing data for solving a problem, and shortens the time to reach an optimal solution.

"Time" is one of the major challenges in *petroleum production optimization* because each execution of the model by a simulator (like Eclipse) takes a long time and the large number of parameters of such problems in a real condition requires the reservoir model to be executed thousands of times by the simulator. It is practically impossible to reach optimal solution in some reservoirs with a large number of grids. Therefore, it seemed necessary to develop an algorithm, which provided a better solution in lower NFE.

A new metaheuristic usually utilizes a new metaphor as the search directory. In this article, the SWA algorithm is introduced as a population-based method for solving optimization problems. One of the advantages of this method over others is that it uses two contradictory types of answers: it uses the worst and the best answers to reach the optimum point. The whale's life style was used as a model in creating this algorithm. Therefore, this animal is described first.

## 2. Sperm whale related background

The sperm whale belongs to the Odontocete suborder of the Class Mammalia, is a toothed whale, and the largest predator. Its senses of taste and smell are weak [33], but its sense of hearing is so strong that whales can use it when communicating with one another. The sperm whale emits sounds to inform other members of the group that prey is nearby. Its eyes cannot roll in the eye sockets and, hence, the eyes are not very strong, but sperm whales can retract and protrude their eyes due to the presence of thick retractor muscles attached around the eyes [34, 35] . This helps them to catch squids [36]. They take approximately 1000 kg each day [37]. Sperm whales have an average length of 16 meters, mean weight of about 45 tons, and the largest brain (weighing up to 18 kilograms) among all the creatures living on earth. Sperm whales feed on squids that live deep in the water. To catch squids, they must go down to the depths of the water (2000-3000 meters deep) while they need to come to the surface to breathe [38]. That is why they experience two opposite poles of their environment in each cycle of breathing and feeding: the surface and, usually, the bottom of the sea. It is believed that sperm whales can stop breathing for up to 90 minutes [39].

Sperm whales usually travel in groups of 6 to 9 and the males and females live in the same group. Moreover, the males may also form weak all-male groups of their own. Of course, most males live alone except for the mating season. Their mating pattern is like this: males fight each other and the final superior male mates with several female whales [40]. Other animals such as orcas (killer whales) may catch weak sperm whales (such as the kids and the females), but male sperm whales are not caught by any animal [41].

## 3. Sperm whale Algorithm

In this research, life style of sperm whales described above was mathematically modeled to introduce a new efficient algorithm called SWA. In this algorithm, each answer represents a sperm whale. Taking the formation of social units of sperm whales into consideration, m×n number of answers were first created, evaluated, and ordered as the initial population. This ordered population was then divided into n Temporary Sub-Groups (TSG) each with m members, and one member was randomly selected from each of the temporary subgroups for every Main Sub-Group (MSG). Figure 1 shows the grouping process. This process, which is the simulation of sperm whale grouping and both male and female exist in each group, somewhat ensures the distribution of solutions in different groups and prevents early local stuck.

Then the following operations were carried out on each of the main subgroups:

1. Each sperm whale experiences two opposite places in its breathing-feeding cycle (it has to come to the surface to breathe and go down to the seabed to feed). Therefore, for each whale, one answer for its current place and another answer for its mirror place were considered, and the objective functions for both points were estimated. However, the problem here was that the mirror reflection of the best answer did not help in finding the optimum answer and only increased the time needed for finding it. That was why only the worst answer was reflected. In the case where the problem had constraints, the reflection to the center of the search space might transfer the point out of the desired space. It was for this reason, and considering the information exchange between whales, that the worst answer was transferred to a random point on the spatial line connecting the worst and the best answers. The best and the worst whale in each group were called the $X_{best}$ and the $X_{worst}$, respectively. So:

$$X_{center} = X_{worst} + c \times X_{best} \quad \ldots\ldots \quad (1)$$

$$X_{reflex} = X_{worst} + 2 \times \left(X_{center} - X_{worst}\right) = 2X_{center} - X_{worst} \quad \ldots\ldots \quad (2)$$

In the above equation, $X_{center}$ is the reflection center and $X_{reflex}$ is the obtained result from the reflection of the worst answer to the reflection point. Moreover, c is called center factor that could be any number.

If $X_{reflex}$ is located outside of the search space, c should decrease in this way: $c = r \times c_i$ which $c_i$ is initial center factor and r is contraction coefficient that is a value less than 1. r and $c_i$ should be set as algorithm parameters. Figure 2 shows the concept of equation 1 and 2. In the case of bound-constrained global optimization problems, the range of the parameter c can be selected in a way that $X_{reflex}$ does not fall beyond the search space range. Let C be a 1×n vector, with n being the number of decision variables. In this case, we can write:

$$\begin{aligned}
&\text{for } i = 1:n \\
&X_{min}(i) \le X_{reflex}(i) \le X_{max}(i) \xrightarrow[\text{Equation (2)}]{} X_{min}(i) \le 2X_{center}(i) - X_{worst}(i) \le X_{max}(i) \\
&\xrightarrow[\text{Equation (1)}]{} X_{min}(i) \le 2\times(X_{worst}(i) + C(i)\times X_{best}(i)) - X_{worst}(i) \le X_{max}(i) \rightarrow \qquad \ldots\ldots (3) \\
&\rightarrow X_{min}(i) \le X_{worst}(i) + 2C(i)\times X_{best}(i) \le X_{max}(i) \rightarrow \frac{X_{min}(i) - X_{worst}(i)}{2X_{best}(i)} \le C(i) \le \frac{X_{max}(i) - X_{worst}(i)}{2X_{best}(i)}
\end{aligned}$$

$c = \text{rand} \times \text{Min}(C)$ …… (4)

rand is a function in MATLAB® that returns a pseudorandom scalar drawn from the standard uniform distribution on the open interval (0,1). In this case, there is no need to examine whether $X_{reflex}$ falls inside the range or not. This increases the speed of obtaining a solution. This also reduces the number of the parameters that need to be tuned, and there would be no need to set r and c.

2. In the next step, a number of whales (q members) with better objective functions are selected from each main subgroup as the Good Gang and, considering the whales' limited vision, a local search is made around the members of this group. This local search is carried out in this way: the answers change in a definable radius (with Q iterations), and if they improved the answer, the improved changed answers will replace it in the Good Gang group.

3. Afterwards, considering the mating behavior of whales (the strongest male mates with several females), a crossover occurs between the best answer of the Good Gang and the rest of the answers in the main subgroup, and between two generated children, one (randomly) takes the place of the mother answer. Finally, after this process is iterated a specified number of times, all answers of the subgroups are placed beside each other in the sperm whales group and are ordered. These processes are repeated until the optimum answer is found. Figure 3 shows the complete flowchart of the algorithm:

In the flowchart of the SWA algorithm, no condition was introduced for considering the constraints. There are various methods to satisfy the constraints: Static Penalty, Dynamic Penalty, Deb's heuristic constrained handling method [42]. A method similar to Deb's Method is considered for the algorithm. Deb is a constraint handling method based on penalty function technique, which needs no specific setting for its performance. If penalty function technique is used, fitness function will be as follows:

$$F(\vec{x}) = f(\vec{x}) + \sum_{i=1}^{N} R_i \left\langle g_i(\vec{x}) \right\rangle^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.. (5)$$

$f(\vec{x})$ : objective function

$\left\langle g_i(\vec{x}) \right\rangle$: constraint violation

$R_i$ : penalty parameter of the i-th inequality constraints

$R_i$ aims at converting constraint violation magnitude into an objective function scale. Consequently, N penalty parameters should be set correctly. In case a problem includes an equality constraint, N will also increase as much as the number of equalities and setting this parameter will be a difficult task. Therefore, Deb method is used. In this method, one of the following processes will happen by two-by-two comparison of the responses.

1) When two feasible solutions are compared, the better response is selected. 2) When a feasible solution and an infeasible solution are compared, the feasible response is selected. 3) Finally, when two infeasible solutions are compared, the response with lower violation is selected. Since the solutions of both objective function and constraint violation terms are compared simultaneously, there will be no need for the penalty parameter term. Of course, realization of the third scenario is prevented to improve the performance of this method in real problems. To investigate a response, its feasibility is checked at first. If it is feasible, its related objective function is calculated. This method somewhat increased the time needed to find the optimum answer.

## 4. Experimental results and Discussion

The performance of this algorithm is discussed from three perspectives.

- Case 1: Its performance in solving 26 benchmarks regardless of Number of Function Evaluations (NFE)
- Case 2: Its performance in solving 3 benchmarks in different dimensions considering NFE
- Case 3: Its performance in solving production optimization

### 4.1. Case 1: twenty six benchmark function

26 benchmark functions (table 1) [43] were solved using this algorithm and its performance was compared with other optimization algorithms such as GA, PSO, DE, BA and the hybrid PSO-BA (Particle Bee Algorithm [PBA]) [43]. All of the problems were solved by Cheng and Lien [43].

In their work the maximum Number of Function Evaluations (NFE) was set to 500000 and any value less than $10^{-12}$ reported as 0. All of the problems are bound-constrained and there is no need to tune c and r for SWA.

The benchmark set includes various functions. A function is called unimodal if it has only one local optimum. In this case, finding the optimal point can be difficult (due to having a flat surface and not providing information to direct the research toward the optimal point). A function is called multimodal if it has more than one local optimum. These functions are used to test the ability of the algorithm to escape from local minima. The p-variable separable function is a function that can be expressed as the sum of p functions with one variable. Finally, Non-separable functions cannot be expressed in this form because of interrelation among their variables [44]. The mean and standard deviation are obtained from 30 independent runs. Table 2 shows the algorithm parameters in all 3 cases. Results are presented in Table 3. The two last rows show the score and rank of the algorithms. Lower scores indicate higher ranks of an algorithm. As can be seen in Table 3, the SWA, PBA, DE, BA, PSO and GA algorithms are respectively the best algorithms. The interesting point is the extremely poor performance of GA compared to the other algorithms. This can be attributed to not properly tuning its parameters. SWA has the best rank with a slight difference compared to PBA; however, its lower number of parameters is a huge advantage over PBA.

### 4.2. Case 2: three benchmark function with different dimensions

The search space dimension is an important parameter in a problem [45]. Functions 1, 2 and 3 of table 1 were solved in different dimensions to better investigate the algorithm parameters. This was done to demonstrate the effects of an increased dimension in a particular type of function. Detailed results of the function 1 ($f_1(x)$) are presented, but the results of the other functions are in a summarized form. $f_1(x)$ three-dimensional diagram can be seen in figure 4.

An initial population of 20 whales was considered to solve the problem. They were divided into four groups each with five members, and the algorithm was executed by assuming the $c_i$ to be 2, q=2, and

Q=10. Here, in order to examine all parameters, c and r were also tuned to show the effects of all parameters.

Figure 5 (a-f) shows the various stages of the algorithm. As is shown in the figure, most solutions reached the global optimum point even in the 10th iteration, with all of them reaching it in the 14th iteration. In fact, the 10th to the 14th iterations serve the purpose of attaining the desired accuracy (which was 0.001 here). Moreover, 50 independent runs were executed for the problem to examine the stability of the algorithm. The results, which confirmed algorithm stability, are presented in Figure 6.

Furthermore, to further examine the ability of this algorithm in solving problems with more dimensions, and to find out how its parameters (c, r, q, and Q) should be set, the procedure explained for 2D was applied for two other dimensions (5D and 10D) and two other functions ($f_2$ and $f_3$). Results were compared with those of two common metaheuristic algorithms (GA and PSO).

Since the random initial population created for each algorithm influences its performance, each algorithm was executed 50 times and their mean results were used for comparison. Results showed the initial $c_i$=2 with the contraction coefficient of 0.9 were suitable values for reaching the final answer. Moreover, in simpler problems, increasing the values of q and Q (parameters that, in fact, control local search) helped in finding the final answer. However, in difficult problems like $f_2$ and $f_3$, increasing the number of crossovers and the number of subgroups helped to reach the final answer. Table 4 shows the controlling parameters of SWA, GA and PSO. Finally, the performance of the algorithm was compared with those of GA and PSO and the results are presented in the table 5.

These results are the means of 50 runs, with 0.001 degree of accuracy for reaching the optimum answer. As shown in the table, the PSO performed better in only one case, but the SWA was superior to PSO and GA in all other cases. The existence of equilibrium between the global search and the local search in the proposed algorithm is one of the features that distinguish it from other algorithms.

Another feature of this algorithm is that, compared to GA and PSO, it finds the optimum answer with a considerably smaller number of NFEs (Number of Function Evaluations), and this greatly helps to find the optimums in optimization problems, because calculating objective functions is usually very time-consuming in this kind of problems. That is why the performance of this algorithm in solving a production optimization problem was investigated.

### 4.3. Case 3: Solving a production optimization problem

This problem is a non-separable multimodal problem that includes many decision variables. A major challenge in solving such problems is the considerable amount of time required for evaluating the objective function (in some problems, an evaluation may even take several hours).

A synthetic reservoir with 12*150*150 grids each with the dimensions of 20*100*100 feet was built. Three horizontal wells had been drilled in this reservoir, with the middle one being the production well and the other two being the injection wells. Permeability in x and y directions was assumed the same. It was 30 md for one-half of the reservoir and 15 md for the other half and 3 md in the vertical direction. In fact, the k values between the injection wells and the production well were 30 for one of the injection well and 15 for the other. The k values were thus selected so that the two injection wells had different conditions. Figure 7 shows details of the problem.

Porosity was assumed 0. 2 in all parts of the reservoir. The purpose of optimization was to maximize net present value for six years by adjusting the parameters of the production rate in the production well and of the injection rates in the injection wells. The parameters were controlled once every six months and, therefore, there were 36 decision variables. Eclipse program was used to simulate the reservoir. Eclipse is a reservoir simulation program and reservoir model is given to it using a file called “Data file”. “Data file” comprises different parts including RUNSPEC, GRID, PROPS, SOLUTION, SUMMARY, and SCHEDULE. In optimization problems, all sections are fixed and only Schedule is changed.

First, a Schedule file was written using the MATLAB program. The MATLAB program then executed the related “Data File” (which included the Schedule file) by calling the Eclipse Parallel program (parallel was used to decrease the simulation run time) and at the end of each run results were stored in the RSM file (Run Summary file). It then read the cumulative quantities of oil and water produced and the cumulative water injection from this file, calculated the NPV (Net Present Value) quantity with the formula below as the value of the objective function, and repeated this process until the last stage of optimization. This optimization problem is modeled as below:

$$max\ f(u^n) \qquad (6)$$

$$f(u^n) = NPV(u^n) = r_o Q_o(u^n) - r_w Q_w(u^n) - c_{ri\,p} Q_{wi}(u^n) \qquad (7)$$

Subject to:

$$Q_o = \sum_{i=1}^{n}\int_0^t q_o dt,\ Q_w = \sum_{i=1}^{n}\int_0^t (WOR * q_o)dt,\ Q_{wi} = \sum_{i=1}^{n}\int_0^t (q_{inj})dt \qquad (8)$$

$$u^n = (q_o^0, q_o^1, ..., q_o^{12}, q_{w1}^0, q_{w1}^1, ..., q_{w1}^{12}, q_{w2}^0, q_{w2}^1, ..., q_{w2}^{12})$$
$$\text{Max}.Water\ Cut = 0.5$$
$$500 < q_o, q_{inj} < 20000$$

$r_o$, $r_w$ and $r_{wi}$ are the price of oil, the cost of water separation and water injection and in a simple case, they assumed \$70 and \$1 and \$5 for each bbl, respectively. $Q_o$, $Q_w$, and $Q_{wi}$ are respectively the accumulative value for produced oil, produced water, and injected water. $q_o$, $q_{inj}$, , and WOR are respectively the daily rate of oil production, water injection, and water to oil ratio in the produced fluid. Maximum acceptable water cut in producing oil is assumed as 0.5 and the well will be shut if the water cut reaches this value.

This problem was solved using the PSO, GA, and SWA algorithms and their performance was compared. Since the precise solution of the problem was not available, two cases were considered. In case 1, each algorithm was executed until a value was reached for NPV that did not change in the following 1000 NFEs. In case 2, each algorithm was executed to a fixed 2000 NFEs. Due to the simplicity of this problem, all 3 algorithms can handle it and they reach the final NPV around \$1600 million and this shows that the global optimum of this problem is around this point.

The parameters were the same as those in the benchmark problems with the exception that the initial population was raised to 100. Figure 8 shows the steps involved in solving the problem with all three algorithms. In case 1, GA reaches NPV=1558 in NFE=9739, PSO reaches NPV=1612 in NFE=7312 and SWA reaches NPV=1598 in NFE=5558. In case 2, GA reaches NPV=1125, PSO reaches NPV=1038, SWA reaches NPV=1315.

So, the SWA algorithm needed fewer NFEs to reach the answer compared to the PSO and GA, or presented better answers at a fixed number of NFEs. Also, PSO performance is better than GA that this is because of that usually PSO works better in continuous search space [46].

The fewer number of required NFEs, in fact, shortens the desired execution by Eclipse. Actually, in production optimization problems, execution of the file by Eclipse takes a very long time. The SWA algorithm, with its greater efficiency, and because of the smaller number of NFEs it needs, considerably reduces the time required for performing the optimization. Also, combination local optimizer with these algorithms can improve the performance of them as a work was done by Ghaedi et al.[47].

## 5. Conclusions

In this study, which was inspired by sperm whale's lifestyle, a new algorithm called SWA was introduced. It was used to solve 26 benchmark functions, 3 benchmark functions in several dimensions and one production optimization and its performance was compared with GA and PSO. Results of experiments summarized as follows:

- SWA needed fewer NFEs to solve most of the problems compared to GA and PSO. This is very important in production optimization (which is a time-consuming process).
- It was found that the SWA could reduce the time required for finding the optimum answer of production optimization by half.
- One of the advantages of SWA over other algorithms is that it uses two contradictory types of answers: it uses the worst and the best answers to reach the optimum point

The problems considered in this article were continuous search problems. The continuation of this research project will be to adjust the SWA algorithm for use in discrete space, because this can reduce the size of the search space and can accelerate the process of finding the optimum answer.

## References List

1. Storn, R. and K. Price, *Differential Evolution – A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces*. 1995: Berkley.
2. Rangaiah, G.P., *Stochastic global optimization: techniques and applications in chemical engineering*. Vol. 2. 2010: World Scientific.
3. Boussaïd, I., J. Lepagnot, and P. Siarry, *A survey on optimization metaheuristics.* Inf. Sci., 2013. **237**: p. 82-117.
4. Kirkpatrick, S., C.D. Gelatt Jr., and M.P. Vecchi, *Optimization by simulated annealing.* Science, 1983. **220**(4598): p. 671–680.
5. Glover, F., *Future paths for integer programming and links to artificial intelligence.* Comp. Operat. Res. , 1986. **13**(5): p. 533–549.
6. Lourenço, H.R., O. Martin, and T. Stützle, *Iterated Local Search: Framework and Applications*, in *Handbook of Metaheuristics*. 2010, Kluwer Academic Publishers: International Series in Operations Research & Management Science. p. 363–397.
7. Alsheddy, A., *Empowerment scheduling: a multi-objective optimization approach using guided local search*, in *School of Computer Science and Electronic Engineering*. 2011, University of Essex.
8. Hooke, R. and T.A. Jeeves, *Direct search'' solution of numerical and statistical problems.* J. Assoc. Comput., 1961. **8**(2): p. 212–229.
9. Rastrigin, L.A., *The convergence of the random search method in the extremal control of a many parameter system.* Autom. Rem. Control 1963. **24**(10): p. 1337–1342.
10. Hansen, P., N. Mladenovic, and J.A.M. Perez, *Variable neighbourhood search: methods and applications.* Ann. Oper. Res., 201. **175**: p. 367–407.
11. Dogan, B. and T. Ölmez, *A new metaheuristic for numerical function optimization: Vortex Search algorithm.* Information Sciences, 2015. **293**: p. 125-145.
12. Golberg, D.E., *Genetic algorithms in search, optimization, and machine learning.* Addion wesley, 1989. **1989**.
13. Kenndy, J. and R. Eberhart. *Particle swarm optimization*. in *Proceedings of IEEE International Conference on Neural Networks*. 1995.
14. Storn, R. and K. Price, *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,*. J. Global Optim. , 1997. **11**: p. 341–359.
15. Larrañaga, P. and J.A. Lozano, *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. 2002, Boston: Kluwer Academic Publishers.
16. Dorigo, M. and M. Birattari, *Ant colony optimization*, in *Encyclopedia of Machine Learning*. 2010, Springer. p. 36-39.
17. Basturk, B. and D. Karaboga. *An artificial bee colony (ABC) algorithm for numeric function optimization*. in *IEEE swarm intelligence symposium*. 2006.
18. Karaboga, D., *An idea based on honey bee swarm for numerical optimization*. 2005, Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
19. Karaboga, D. and B. Basturk, *A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm.* J. Global Optim. , 2007. **39**(3): p. 459-471.

20. Karaboga, D. and B. Basturk, *On the performance of artificial bee colony (abc) algorithm.* Appl. Soft Comput., 2008. **8** (1): p. 687–697.
21. Geem, Z.W., J.H. Kim, and G. Loganathan, *A new heuristic optimization algorithm: harmony search.* Simulation, 2001. **76**(2): p. 60-68.
22. Passino, K.M., *Biomimicry of bacterial foraging for distributed optimization and control.* Control Systems, IEEE, 2002. **22**(3): p. 52-67.
23. Kashan, A.H. *League championship algorithm: a new algorithm for numerical function optimization*. in *Soft Computing and Pattern Recognition, 2009. SOCPAR'09. International Conference of*. 2009. IEEE.
24. Yang, X.-S., *Nature-Inspired Metaheuristic Algorithms*. 2008, UK: Luniver Press.
25. Yang, X.-S. *Firefly algorithms for multimodal optimisation*. in *5th Symposium on Stochastic Algorithms, Foundations and Applications*. 2009.
26. He, S., Q.H. Wu, and J. Saunders, *Group search optimizer: an optimization algorithm inspired by animal searching behavior.* Evolutionary Computation, IEEE Transactions on, 2009. **13**(5): p. 973-990.
27. Gandomi, A.H., X.-S. Yang, and A.H. Alavi, *Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems.* Engineering with Computers 2011. **29**: p. 17-35.
28. Gandomi, A.H. and A.H. Alavi, *Krill herd: a new bio-inspired optimization algorithm.* Communications in Nonlinear Science and Numerical Simulation, 2012. **17**(12): p. 4831-4845.
29. Alatas, B., *A novel chemistry based metaheuristic optimization method for mining of classification rules.* Expert Systems with Applications, 2012. **39**: p. 11080–11088.
30. Salimi, H., *Stochastic Fractal Search: A Powerful Metaheuristic Algorithm.* Knowledge-Based Systems, 2014. **75**: p. 1-18.
31. Cheng, M.-Y. and D. Prayogo, *Symbiotic Organisms Search: A new metaheuristic optimization algorithm.* Computers and Structures, 2014. **139**: p. 198-112.
32. Husseinzadeh Kashan, A., *A new metaheuristic for optimization: Optics inspired optimization (OIO).* Computers & Operations Research, 2015. **55**: p. 99-125.
33. Oelschläger, H.H. and B. Kemp, *Ontogenesis of the sperm whale brain.* Journal of Comparative Neurology, 1998. **399**(2): p. 210-228.
34. Bjergager, P., S. Heegard, and J. Tougaard, *Anatomy of the eye of the sperm whale (Physeter macrocephalus L.).* Aquatic Mammals, 2003(29): p. 31-36.
35. Clarke, M.R., H.R. Martins, and P. Pascoe, *The diet of sperm whales (Physeter macrocephalus Linnaeus 1758) off the Azores.* Phil. Trans. R. Soc. Lond. , 1993: p. 67–82.
36. Fristrup, K.M. and G.R. Harbison, *How do sperm whales catch squids?* Marine Mammal Science, 2002. **18**(1): p. 42-54.
37. Lockyer, C., *Estimates of growth and energy budget for the sperm whale* in *In Mammals in the Seas*. 1981, Physeter catodon.
38. Lee, J.J. *Elusive Whales Set New Record for Depth and Length of Dives Among Mammals*. 2014; Available from: http://news.nationalgeographic.com/news/2014/03/140326-cuvier-beaked-whale-record-dive-depth-ocean-animal-science/.
39. Perrin, W.F., B. Wursig, and J. Thewissen, *Encyclopedia of marine mammals*. 2009: Academic Press.

40. Whitehead, H. and L. Weilgart, *The sperm whale: social females and roving males.* Cetacean societies: field studies of dolphins and whales, 2000: p. 154-172.
41. Pitman, R.L., et al., *Killer whale predation on sperm whales: observations and implications.* Marine mammal science, 2001. **17**(3): p. 494-507.
42. Deb, K., *An efficient constraint handling method for genetic algorithms.* Computer methods in applied mechanics and engineering, 2000. **186**(2): p. 311-338.
43. Cheng, M.-Y. and L.-C. Lien, *Hybrid artificial intelligence-based PBA for benchmark functions and facility layout design optimization.* J Comput Civil Eng, 2012 **26**: p. 612–624.
44. Karaboga, D. and B. Akay, *A comparative study of Artificial Bee Colony algorithm.* Applied Mathematics and Computation, 2009. **214**: p. 108-132.
45. Boyer, D.O., C.H. Martfnez, and N.G. Pedrajas, *Crossover operator for evolutionary algorithms based on population features.* Journal of Artificial Intelligence Research, 2005. **24**: p. 1–48.
46. Ebrahimi, A. and E. Khamehchi, *The Use of Optimization Procedures to Estimate Minimum Miscibility Pressure.* Petroleum Science and Technology, 2014. **32**(8): p. 947-957.
47. Ghaedi, M., A. Nejad Ebrahimi, and a.M.R. Pishvaie, *Application of Genetic Algorithm for Optimization of Separator Pressures in Multistage Production Units.* Chemical Engineering Communications, 2014. **201**(7): p. 926-938.

## Table Captions

Table 1- The detailed of benchmark functions (D: Dimensions, U: Unimodal, M: Multimodal, N: Non-separable, S: Separable).

Table 2- optimization algorithms parameters used in solving 26 benchmarks (case 1)

Table 3- optimization algorithms performance comparison on benchmark functions (case 1)

Table 4- Controlling parameter values of SWA, GA and PSO in solving 3 benchmarks in different dimensions (case 2)

Table 5- Results of SWA vs. GA and PSO in solving 3 benchmarks in different dimensions (case 2)

## Figure Captions

Figure 1- Grouping process in SWA Algorithm

Figure 2- reflection of worse answer in SWA

Figure 3- SWA flowchart

Figure 4- 3D Diagram of Rastrigin's function ($f_1(x)$)

Figure 5- location of whales on 2D benchmark f1 in a) Initial population b) 2nd iteration c) 4th iteration d) 6th iteration e) 10th iteration f) 14th iteration

Figure 6- SWA stability plot for 50 individual runs on benchmark f1

Figure 7- 3D view of the reservoir in production optimization problem (case 3)

Figure 8- comparison of SWA performance with GA and PSO in solving production optimization (case 3)

Table 1- The detailed of benchmark functions (D: Dimensions, U: Unimodal, M: Multimodal, N: Non-separable, S: Separable).

| No | Name | Range | D | Type | Formulation | Min |
|---|---|---|---|---|---|---|
| 1 | Rastrigin | [-5.12,5.12] | n | MS | $f_1(x)=10n+\sum_{i=1}^{n}[x_i{}^2-10\cos(2\pi x_i)]$ | 0 |
| 2 | De Jong (Sphere) | [-5.12,5.12] | n | US | $f_2(x)=\sum_{i=1}^{n}x_i{}^2$ | 0 |
| 3 | Griewank | [-600,600] | n | MN | $f_3(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i{}^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$ | 0 |
| 4 | Beale | [-4.5,4.5] | 2 | UN | $f_4(x)=(1.5-x_1+x_1x_2)^2+(2.25-x_1+x_1x_2^2)^2+\ldots$ $\ldots(2.625-x_1+x_1x_2^3)^2$ | 0 |
| 5 | Easom | [-100,100] | 2 | UN | $f_5(x)=-\cos(x_1)\cos(x_2)\exp(-(x_1-\pi)^2-(x_2-\pi)^2)$ | -1 |
| 6 | Matyas | [-10,10] | 2 | UN | $f_6(x)=0.26(x_1^2+x_2^2)-0.48x_1x_2$ | 0 |
| 7 | Bohachevsky1 | [-100,100] | 2 | MS | $f_7(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)-0.4\cos(4\pi x_2)+0.7$ | 0 |
| 8 | Booth | [-10,10] | 2 | MS | $f_8(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$ | 0 |
| 9 | Michalewicz2 | [0, $\pi$] | 2 | MS | $f_9(x)=-\sum_{i=1}^{D}\sin(x_i)(\sin(ix_i^2/\pi))^{20}$ | -1.8013 |
| 10 | Schaffer | [-100,100] | 2 | MN | $f_{10}(x)=0.5+\frac{\sin^2(\sqrt{x_1^2+x_2^2})-0.5}{(1+0.001(x_1^2+x_2^2))^2}$ | 0 |
| 11 | Six Hump Camel Back | [-5,5] | 2 | MN | $f_{10}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$ | -1.03163 |
| 12 | Boachevsky2 | [-100,100] | 2 | MN | $f_{12}(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1)(4\pi x_2)+0.3$ | 0 |
| 13 | Boachevsky3 | [-100,100] | 2 | MN | $f_{13}(x)=x_1^2+2x_2^2-0.3\cos(3\pi x_1+4\pi x_2)+0.3$ | 0 |
| 14 | Shubert | [-10,10] | 2 | MN | $f_{14}(x)=(\sum_{i=1}^{5}i\cos(i+1)x_1+i)(\sum_{i=1}^{5}i\cos((i+1)x_2+i))$ | -186.73 |
| 15 | Colville | [-10,10] | 4 | UN | $f_{15}(x)=100(x_1^2-x_2)^2+(x_1-1)^2+(x_3-1)^2+90(x_3^2-x_4)^2+..$ $..10.1(x_2-1)^2+(x_4-1)^2+19.8(x_2-1)(x_4-1)$ | 0 |
| 16 | Michalewicz5 | [0, $\pi$] | 5 | MS | $f_{16}(x)=-\sum_{i=1}^{D}\sin(x_i)(\sin(ix_i^2/\pi))^{20}$ | -4.6877 |
| 17 | Zakharov | [-5,10] | 10 | UN | $f_{17}(x)=\sum_{i=1}^{D}x_i^2+(\sum_{i=1}^{D}0.5ix_i^2)^2+(\sum_{i=1}^{D}0.5ix_i^2)^4$ | 0 |
| 18 | Michalewicz10 | [0, $\pi$] | 10 | MS | $f_{18}(x)=-\sum_{i=1}^{D}\sin(x_i)(\sin(ix_i^2/\pi))^{20}$ | -9.6602 |
| 19 | Step | [-5.12,5.12] | 30 | US | $f_{19}(x)=\sum_{i=1}^{D}(x_i+0.5)^2$ | 0 |
| 20 | SumSquares | [-10,10] | 30 | US | $f_{20}(x)=\sum_{i=1}^{D}ix_i^2$ | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 21 | Quartic | [-1.28,1.28] | 30 | US | $f_{21}(x)=\sum_{i=1}^{D} i x_i^4 + \text{Rand}$ | 0 |
| 22 | Schwefel 2.22 | [-10,10] | 30 | UN | $f_{22}(x)=\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i=1}^{D}\lvert x_i\rvert$ | 0 |
| 23 | Schwefel 1.2 | [-100,100] | 30 | UN | $f_{23}(x)=\sum_{i=1}^{D}(\sum_{j=1}^{D} x_j)^2$ | 0 |
| 24 | Rosenbrock | [-30,30] | 30 | MN | $f_{24}(x)=\sum_{i=1}^{D}100(x_{i+1}-x_i^2)^2+(x_i-1)^2$ | 0 |
| 25 | Dixon-Price | [-10,10] | 30 | UN | $f_{25}(x)=(x_1-1)^2+\sum_{i=2}^{D} i(2x_i^2-x_i-1)^2$ | 0 |
| 26 | Ackley | [-32,32] | 30 | MN | $f_{26}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{D}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{D}\cos(2\pi x_i))+20+e$ | 0 |

Table 2- optimization algorithms parameters used in solving 26 benchmarks (case 1)

| **Optimization algorithm** | **Parameters** |
|---|---|
| GA | population size=50; mutation rate=0.01; crossover rate=0.8; generation gap=0.9 |
| DE | population size=50; crossover rate=0.9; scaling factor=0.5 |
| PSO | population size=50; inertia weight=0.9-0.7; limit of velocity= $X_{min}$/10- $X_{min}$/10 |
| BA | population size=50; elite bee number=n/2; best bee number=n/4; random bee number=n/4; elite bee neighborhood number=2, best bee neighborhood number=1 |
| PBA | population size=50; elite bee number=n/2; best bee number=n/4; random bee number=n/4; inertia weight=0.9-0.7; limit of velocity= $X_{min}$/10- $X_{min}$/10; PSO iteration of elite bees=15; PSO iteration of best bees=9 |
| SWA | Number of main groups=10; group size=5; good gang size=2; local search iteration=10 |

Table 3- optimization algorithms performance comparison on benchmark functions (case 1)

| f | D | Min | | GA | DE | PSO | BA | PBA | SWA |
|---|---|---|---|---|---|---|---|---|---|
| **f1(x)** | 30 | 0 | Mean | 52.92259 (4) | 11.716728 (2) | 43.9771369 (3) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 4.56486 | 2.538172 | 11.728676 | 0 | 0 | 0 |
| **f2(x)** | 30 | 0 | Mean | 1.11 E+03 (2) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 74.21447 | 0 | 0 | 0 | 0 | 0 |
| **f3(x)** | 30 | 0 | Mean | 10.63346 (5) | 0.00148 (2) | 0.01739 (4) | 0 (1) | 0.00468 (3) | 0 (1) |
| | | | SD | 1.16146 | 0.00296 | 0.02081 | 0 | 0.00672 | 0 |
| **f4(x)** | 2 | 0 | Mean | 0 (1) | 0 (1) | 0 (1) | 1.88E-05 (2) | 0 (1) | 0 (1) |
| | | | SD | 0 | 0 | 0 | 1.94E-05 | 0 | 0 |
| **f5(x)** | 2 | -1 | Mean | -1 (1) | -1 (1) | -1 (1) | -0.99994 (2) | -1 (1) | -1 (1) |
| | | | SD | 0 | 0 | 0 | 4.50E-05 | 0 | 0 |
| **f6(x)** | 2 | 0 | Mean | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 0 | 0 | 0 | 0 | 0 | 0 |
| **f7(x)** | 2 | 0 | Mean | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 0 | 0 | 0 | 0 | 0 | 0 |
| **f8(x)** | 2 | 0 | Mean | 0 (1) | 0 (1) | 0 (1) | 0.00053 (2) | 0 (1) | 0 (1) |
| | | | SD | 0 | 0 | 0 | 0.00074 | 0 | 0 |
| **f9(x)** | 2 | -1.8013 | Mean | -1.8013 (1) | -1.8013 (1) | -1.57287 (2) | -1.8013 (1) | -1.8013 (1) | -1.8013 (1) |
| | | | SD | 0 | 0 | 0.11986 | 0 | 0 | 0 |
| **f10(x)** | 2 | 0 | Mean | 0.00424 (2) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 0.00476 | 0 | 0 | 0 | 0 | 0 |
| **f11(x)** | 2 | -1.0316 | Mean | -1.03163 (1) | -1.03163 (1) | -1.03163 (1) | -1.03163 (1) | -1.03163 (1) | -1.03163 (1) |
| | | | SD | 0 | 0 | 0 | 0 | 0 | 0 |
| **f12(x)** | 2 | 0 | Mean | 0.06829 (2) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 0.07822 | 0 | 0 | 0 | 0 | 0 |
| **f13(x)** | 2 | 0 | Mean | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 0 | 0 | 0 | 0 | 0 | 0 |
| **f14(x)** | 2 | -186.73 | Mean | -186.73 (1) | -186.73 (1) | -186.73 (1) | -186.73 (1) | -186.73 (1) | -186.73 (1) |
| | | | SD | 0 | 0 | 0 | 0 | 0 | 0 |
| **f15(x)** | 4 | 0 | Mean | 0.01494 (3) | 0.04091 (4) | 0 (1) | 1.1176 (5) | 0 (1) | 0.00544 (2) |
| | | | SD | 0.00736 | 0.08198 | 0 | 0.46623 | 0 | 0.00063 |
| **f16(x)** | 5 | -4.6877 | Mean | -4.64483 (3) | -4.68348 (2) | -2.49087 (4) | -4.6877 (1) | -4.6877 (1) | -4.6877 (1) |
| | | | SD | 0.09785 | 0.01253 | 0.25695 | 0 | 0 | 0 |
| **f17(x)** | 10 | 0 | Mean | 0.01336 (2) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 0.00453 | 0 | 0 | 0 | 0 | 0 |
| **f18(x)** | 10 | -9.6602 | Mean | -9.49683 (4) | -9.59115 (3) | -4.00718 (5) | -9.6602 (1) | -9.6602 (1) | -9.61387 (2) |
| | | | SD | 0.14112 | 0.06421 | 0.50263 | 0 | 0 | 0.00236 |
| **f19(x)** | 30 | 0 | Mean | 1.17 E+03 (3) | 0 (1) | 0 (1) | 5.1237 (2) | 0 (1) | 0 (1) |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | SD | 76.56145 | 0 | 0 | 0.39209 | 0 | 0 |
| **f20(x)** | 30 | 0 | Mean | 1.48 E+02 (2) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 12.40929 | 0 | 0 | 0 | 0 | 0 |
| **f21(x)** | 30 | 0 | Mean | 0.1807 (6) | 0.00136 (4) | 0.00116 (3) | 1.72 E-06 (2) | 0.00678 (5) | 0 (1) |
| | | | SD | 0.02712 | 0.00042 | 0.00028 | 1.85E-06 | 0.00133 | 0 |
| **f22(x)** | 30 | 0 | Mean | 11.0214 (3) | 0 (1) | 0 (1) | 0 (1) | 7.59 E-10 (2) | 0 (1) |
| | | | SD | 1.38686 | 0 | 0 | 0 | 7.10E-10 | 0 |
| **f23(x)** | 30 | 0 | Mean | 7.40 E+03 (2) | 0 (1) | 0 (1) | 0 (1) | 0 (1) | 0 (1) |
| | | | SD | 1.14E+03 | 0 | 0 | 0 | 0 | 0 |
| **f24(x)** | 30 | 0 | Mean | 1.96 E+05 (5) | 18.20394 (4) | 15.088617 (3) | 28.834 (5) | 4.2831 (1) | 13.36393 (2) |
| | | | SD | 3.85E+04 | 5.03619 | 24.170196 | 0.10597 | 5.7877 | 4.0295 |
| **f25(x)** | 30 | 0 | Mean | 1.22 E+03 (3) | 0.66667 (2) | 0.66667 (2) | 0.66667 (2) | 0.66667 (2) | 0 (1) |
| | | | SD | 2.66E+02 | E-09 | E-08 | 1.16E-09 | 5.65E-10 | 0 |
| **f26(x)** | 30 | 0 | Mean | 14.67178 (4) | 0 (1) | 0.16462 (3) | 0 (1) | 3.12 E-8 (2) | 0 (1) |
| | | | SD | 0.17814 | 0 | 0.49387 | 0 | 3.98E-08 | 0 |
| **Score** | | | | **64** | **39** | **46** | **40** | **35** | **29** |
| **Final Rank** | | | | **6** | **3** | **5** | **4** | **2** | **1** |

Table 4- Controlling parameter values of SWA, GA and PSO in solving 3 benchmarks in different dimensions (case 2)

| **Algorithm** | **Dimensions** | **Parameters** |
|---|---|---|
| **GA** | 5 | Initial Population=25, pc=0.8 , pm=0.1 , mutation rate= 0.02, Tournament selection |
| | 10 | Initial Population=50, pc=0.8 , pm=0.1 , mutation rate= 0.02, Tournament selection |
| **PSO** | 5 | Initial Population=25, c1=c2=2 , intertia weight=0.75 |
| | 10 | Initial Population=25, c1=c2=2 , intertia weight=0.75 |
| **SWA** | 5 | Initial Population=25,m=n=5, C=2, Cdamp=0.95, q=2, Q=10 |
| | 10 | Initial Population=50,m=10, n=5, C=2, Cdamp=0.95, q=4, Q=10 |

Table 5- Results of SWA vs. GA and PSO in solving 3 benchmarks in different dimensions (case 2)

| **Function** | **Dimensions** | **GA (Number of Function Evaluation)** | **PSO (NFE)** | **SWA (NFE)** |
|---|---|---|---|---|
| **f1** | 5 | 831 | 1543 | 362 |
| **f2** | 5 | 11211 | 6827 | 7510 |
| **f3** | 5 | 15611 | 30022 | 2873 |
| **f1** | 10 | 2112 | 3235 | 895 |
| **f2** | 10 | 25112 | 14569 | 10231 |
| **f3** | 10 | 39239 | 86713 | 8712 |

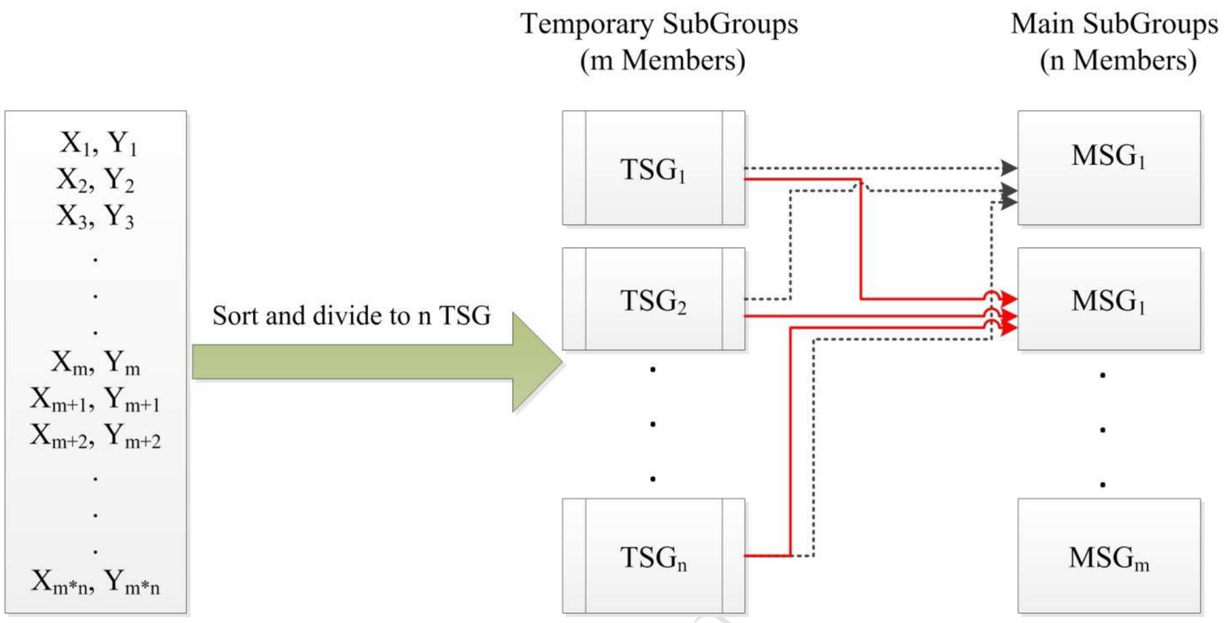

Figure 1- Grouping process in SWA Algorithm

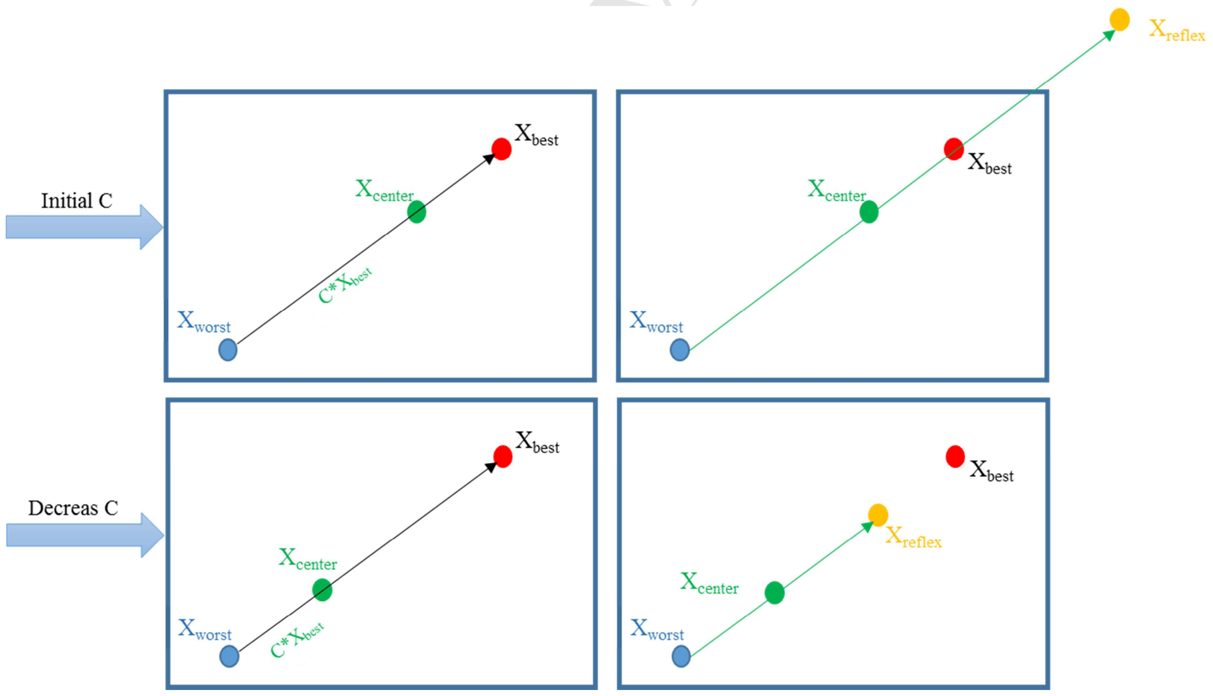

Figure 2- reflection of worse answer in SWA

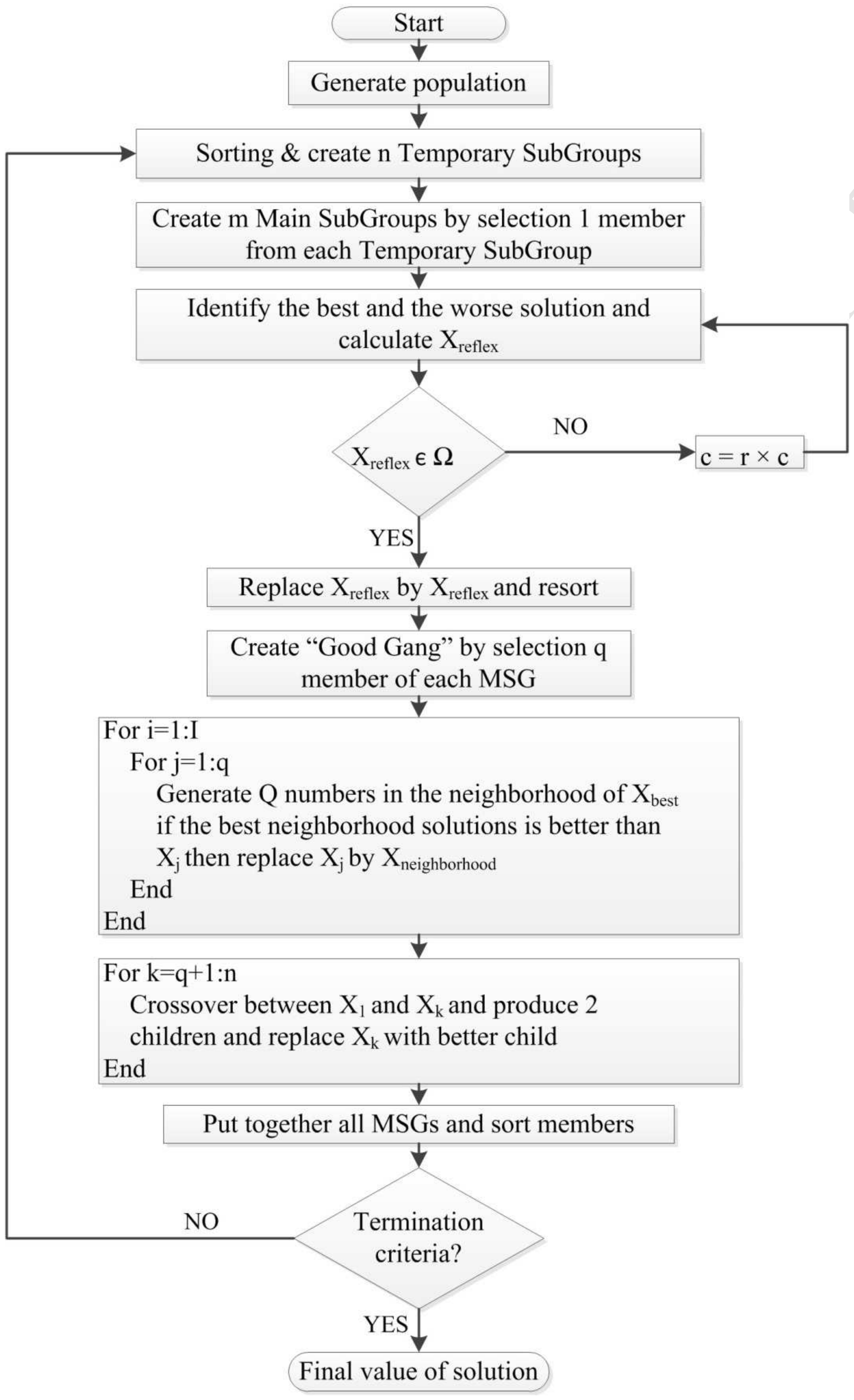
Start
Generate population
Sorting & create n Temporary SubGroups
Create m Main SubGroups by selection 1 member from each Temporary SubGroup
Identify the best and the worse solution and calculate $X_{reflex}$
$X_{reflex} \in \Omega$
NO
$c = r \times c$
YES
Replace $X_{reflex}$ by $X_{reflex}$ and resort
Create "Good Gang" by selection q member of each MSG
For i=1:I
For j=1:q
Generate Q numbers in the neighborhood of $X_{best}$
if the best neighborhood solutions is better than
$X_j$ then replace $X_j$ by $X_{neighborhood}$
End
End
For k=q+1:n
Crossover between $X_1$ and $X_k$ and produce 2
children and replace $X_k$ with better child
End
Put together all MSGs and sort members
Termination criteria?
NO
YES
Final value of solution

Figure 3- SWA flowchart

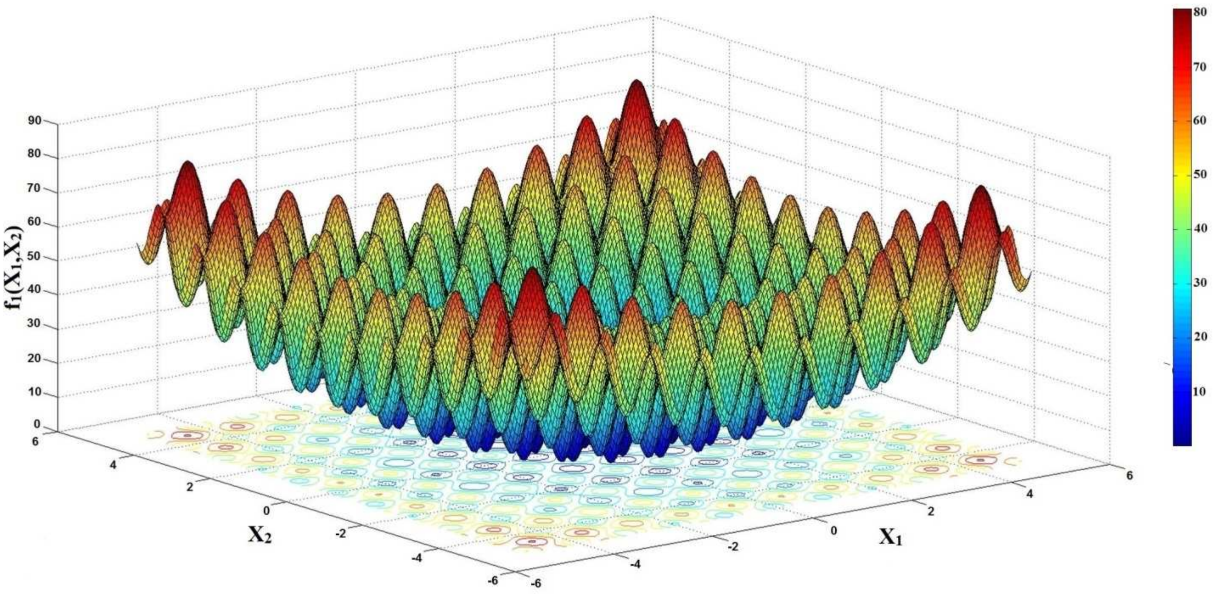

Figure 4- 3D Diagram of Rastrigin’s function ($f_1(x)$)

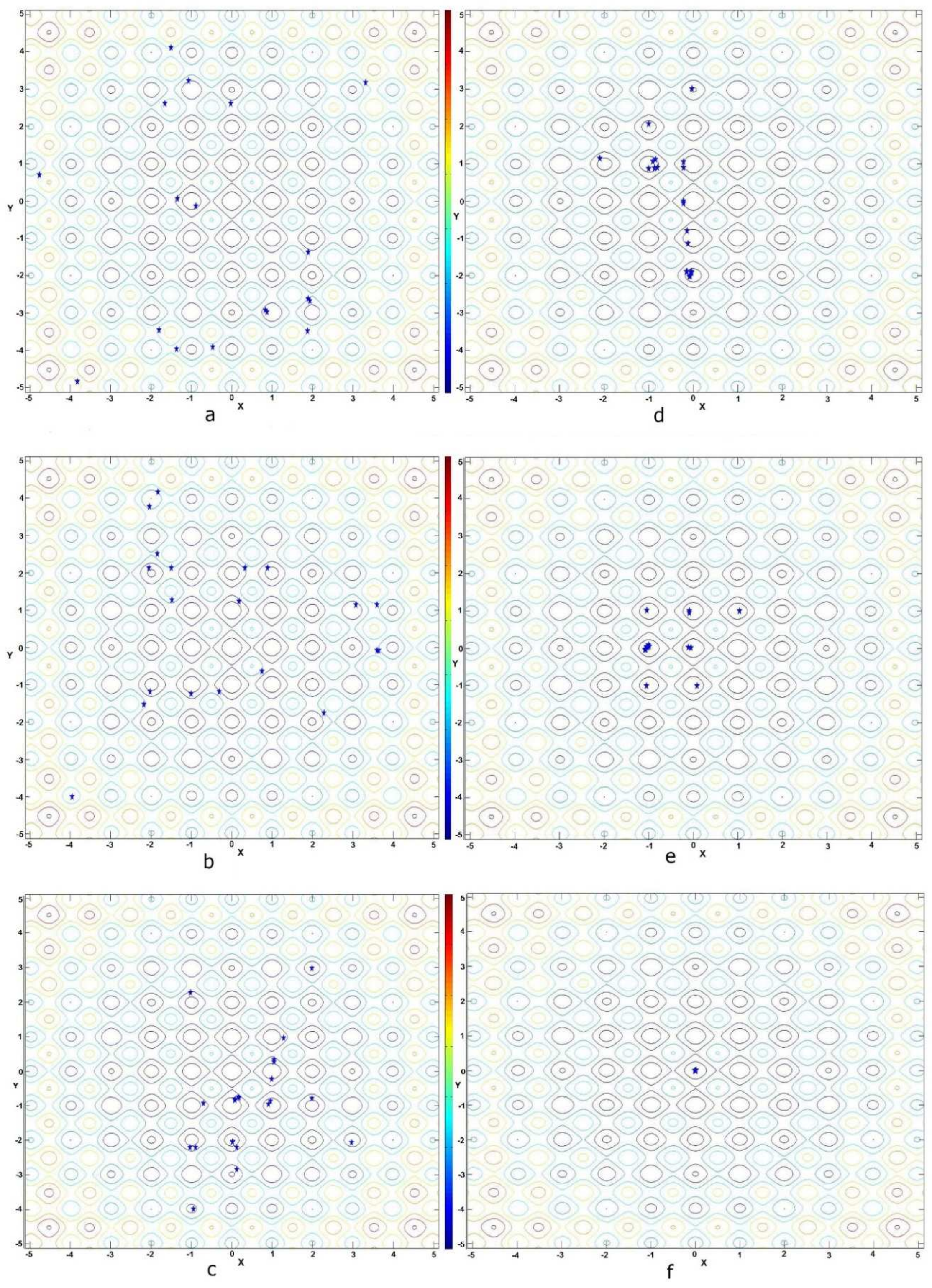

Figure 5- location of whales on 2D benchmark f1 in a) Initial population b) 2nd iteration c) 4th iteration d) 6th iteration e) 10th iteration f) 14th iteration

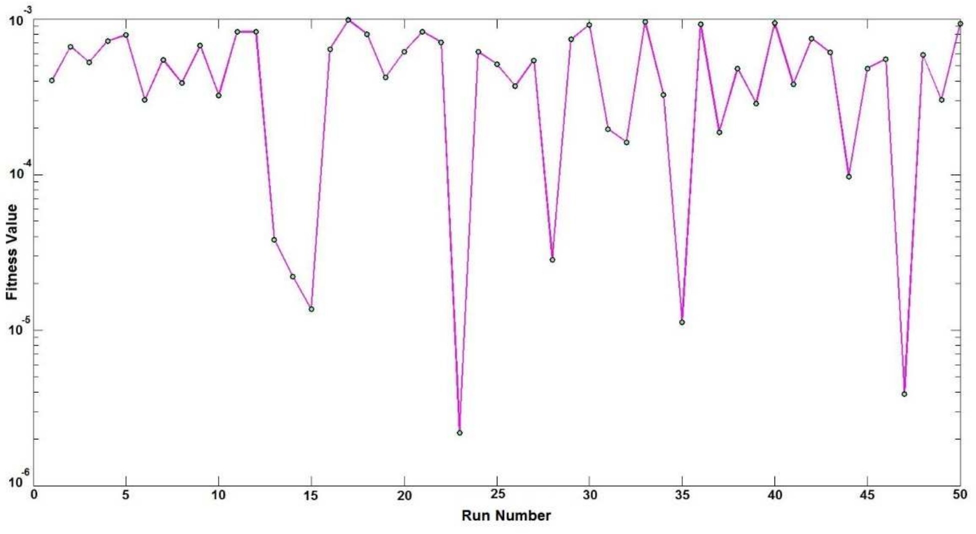

Figure 6- SWA stability plot for 50 individual runs on benchmark f1

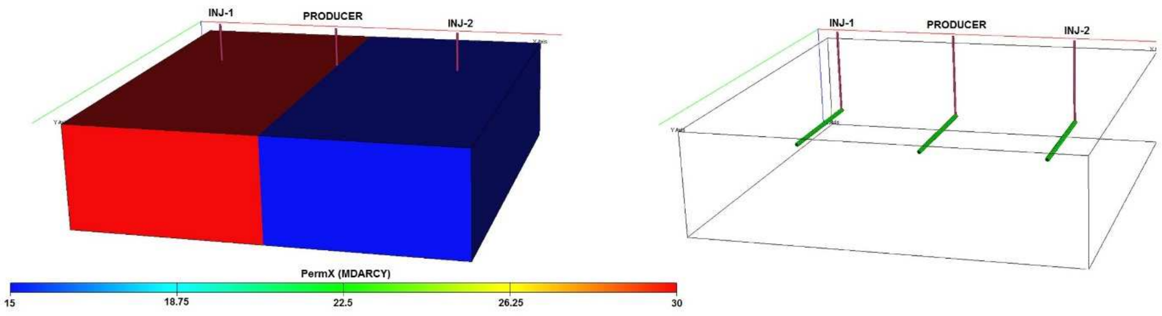

Figure 7- 3D view of the reservoir in production optimization problem (case 3)

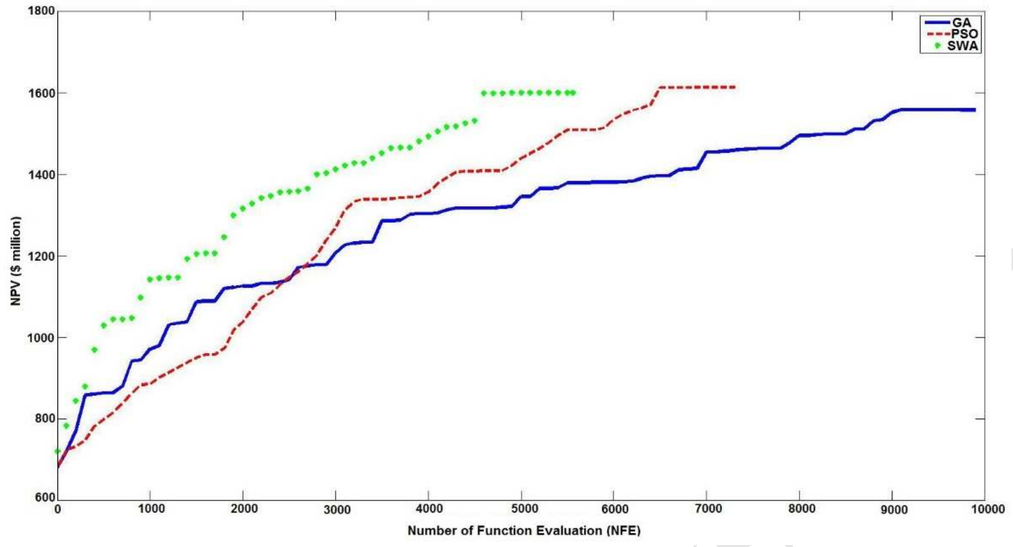

Figure 8- comparison of SWA performance with GA and PSO in solving production optimization (case 3)

## Highlights

- A new algorithm called SWA (Sperm Whale Algorithm) was introduced. The whale's life style was used as a model in creating this algorithm.
- One of the advantages of this method over others is that it uses two contradictory types of answers: it uses the worst and the best answers to reach the optimum point.
- SWA was used to solve 26 benchmark functions, 3 benchmark functions in several dimensions and one production optimization and its performance was compared with GA (Genetic Algorithm) and PSO (Particle Swarm Optimization).
- Results shows that SWA needed fewer NFEs (Number of Function Evaluations) to solve most of the problems compared to GA and PSO. This is very important in production optimization.
- It was found that the SWA could reduce the time required for finding the optimum answer by half.