Guohua Fang, Wei Guo, Xianfeng Huang, Xinyi Si, Fei Yang, Qian Luo, Ke Yan

College of Water Conservancy and Hydropower Engineering, Hohai University, Nanjing, China

A New Multi-objective Optimization Algorithm: MOAFSA and its Application

Abstract. *This paper proposes a multi-objective artificial fish swarm algorithm (MOAFSA), which imitates the behaviors of fish for local search, uses the quick sort method to get non-dominated solution set, and cuts the external set according to the crowding distance. This paper firstly uses MOAFSA for multi-objective functions test. Results show that MOAFSA has a faster convergence speed and the corresponding Pareto set is more evenly distributed; then MOAFSA is applied in the scheduling optimization of hydropower station reservoir.*

Streszczenie. *Przedstawiono nowy algorytm optymalizacyjny MOAFSA (multiobjective artificial fish swarm algorithm) bazujący na ławicy ryb. Algorytm sprawdzono na przykładzie projektowania obciążeń hydroelektrowni. (**Wieloobiektowy algorytm optymalizacyjny MOAFSA i jego zastosowanie**)*

Keywords: multi-objective artificial fish swarm algorithm, quick sort, non-dominated solution set, crowding distance, reservoir scheduling
Słowa kluczowe: optymalizacja, algorytm obptymalizacyjny,

1. Introduction

As a heuristic random search algorithm, evolutionary algorithm (EA) has been successfully applied in the field of engineering optimization, which is combined with solving strategies of non-dominated solution set to produce evolutionary multi-objective optimization algorithm. Early in 1985, Schaffe proposed vector evaluation genetic algorithm (VEGA), which was considered as a pioneering work in which the evolutionary algorithm is used to solve multi-objective optimization problem [1]; in 1990s, a large number of excellent multi-objective evolutionary algorithms emerged; in 1994, Srinivas and Deb proposed Non-Dominated Sorting Genetic Algorithm (NSGA) [2]; afterwards in 2002, Deb improved it and proposed the classic algorithm: NSGA-II [3]; Zitzler and Thiele proposed Strength Pareto Evolutionary Algorithm(SPEA) [4] in 1999; then in 2001, an improved version of SPEA2 [5] was proposed.

Artificial fish swarm algorithm (AFSA), proposed by Li Xiaolei [6] in 2002, is a new optimization algorithm based on the imitation of fish behaviors, which can nicely solve the function optimization and other problems. AFSA has been well applied in different engineering fields: in 2007, Jiang and Wang employed it to solve detection problem [7]; Jiang and Mastorakis use it in the multi-threshold image segmentation [8]; in order to forecast the indices of Shanghai Stock Exchange, Shen and Guo [9] use AFSA to optimize the Radial Basis Function in 2011; Wang extends the normal artificial fish to two-dimension artificial fish to solve problems about optimization and allocation of the workshop's production capability [10].

This paper intends to improve basic AFSA into MOAFSA, tests the feasibility of the algorithm by calculating and analyzing some typical test functions of MOP, and then applies it in reservoir optimal scheduling. The structure of this paper is as follows: in Section II, the basic artificial fish swarm algorithm is introduced and simply analyzed; in Section III, MOAFSA is proposed, and its idea and the structure is expounded; in section IV, MOAFSA is used to solve the classic multi-objective mathematical problems, and is compared with NSGA-II and SPEA2 algorithms, then it is applied in the reservoir scheduling; finally, Section V brings this paper to a conclusion.

2. AFSA

Artificial fish swarm algorithm (AFSA) [11] is to imitate the fish preying, fish swarming and fish following behaviors. This algorithm [6] have the characteristics of artificial fish, and due to the above acts, it is more capable of global optimization. Suppose there are N artificial fish (AF), $d_{ij}=\|X_i-X_j\|$ stands for the distance between two AF individuals, $Step$ is the moving step length, δ is the crowd factor, and $Visual$ represents the visual distance, then $\Omega=\left\{X_j \,\middle|\, \|X_i-X_j\|<Visual\right\}$ is the visual space that current AF can perceive.

2.1 Preying Behavior

Normally AF swims randomly and will go quickly towards the water area where more food is discovered. Suppose current state of AF is X_i and its food concentration (objective function value) is Y_i; select a state X_j randomly in its visual distance. If $Y_i<Y_j$, AF goes forward a step toward X_j, or otherwise, it will reselect another state randomly. If AF is still unsatisfied after Try_number times, it will move a step randomly in its visual space. The preying behavior in Function is as follows:

(1)
$$prey(X_i)=\begin{cases} X_i+Rand(\ )\cdot Step\cdot\dfrac{X_j-X_i}{\|X_j-X_i\|} & if\ (Y_i<Y_j) \\ prey(X_i) & else\end{cases}$$

2.2 Swarming Behavior

It refers to a behavior that AF individuals swim spontaneously to guarantee colony's existence and avoid enemies' invasion. Suppose current state of AF is X_i, the number of AF in its visual space is n_f. Normally AF swims randomly and would go quickly towards the water area where more food is discovered. $if\left(Y_c/n_f>\delta Y_i\right)$, it means the central position X_c is not that crowded, then AF will move a step towards X_c. Otherwise, AF executes preying behavior. The swarming behavior function is shown as follows:

(2)
$$swarm(X_i)=\begin{cases} X_i+Rand(\ )\cdot Step\cdot\dfrac{X_c-X_i}{\|X_c-X_i\|} & if\left(\dfrac{Y_c}{n_f}>\delta Y_i\right) \\ prey(X_i) & else\end{cases}$$

2.3 Following Behavior

It means a behavior that neighboring AFs will trail and reach the food quickly once a single AF has found more food. Suppose X_{max} is the position of AF with the maximum food concentration in the visual space of X_i, $if\left(Y_{max}/n_f > \delta Y_i\right)$, then AF will move a step towards X_{max}. Otherwise, AF executes preying behavior. The following behaviot funtion is as follows:

$$(3)\qquad follow(X_i)=\begin{cases} X_i+Rand(\ )\cdot Step\cdot\dfrac{X_{max}-X_i}{\|X_{max}-X_i\|} & if\left(\dfrac{Y_{max}}{n_f}>\delta Y_i\right)\\ prey(X_i) & else\end{cases}$$

AFSA has global search capacity and strong robustness, and is insensitive to initial values. But this algorithm uses a fixed step and a fixed visual space, which cause the algorithm to run into a "premature" phenomenon at the later stage, and cannot jump out of local optimal solution, bringing great trouble to optimization calculation. Jiang and Yuan suggested using adaptive steps, adaptive visual distance, the recorder array, proposition of the survival and competitive mechanism [12] to improve the algorithm; Jiang and Mastorakis proposed three ways to improve the step [8] and verified it. This paper employs the third way:

$$(4)\qquad Step_{k+1}=\frac{\beta\times\left(N-k\right)}{N}Step_k,\quad \beta\in 1.1\sim 1.5$$

Where: k--current iteration time, N--all iteration times.

3. MOAFSA

3.1 Algorithm idea

MOAFSA improves random search behaviors of AFs and is combined with the sorting method for non-dominated solution set; it uses the improved AF's preying, following, swarming behaviosr to search the optimal solution; constructs the external set to store non-dominant solutions by quick sort method; deletes the redundant individuals from the external set by sequencing their crowding distances so as to maintain the distributivity and diversity of the solution set.

MOEA spends most time in constructing the non-dominated solution set. This paper adopts the construction method NDSet() proposed by Zheng [13], in order to accelerate the algorithm. The crowding distance calculation refers to the Function: Crowding-distance-assignment(P) in Reference [3], and we can get each individual's crowding distance P.dist.

The improved AF's behaviors are as follows:

(1) Improved preying behavior

The condition of whether the preying behavior can be carried out or not is changed from $if\left(Y_i < Y_j\right)$ (the food concentration of X_i is smaller than X_j) into $if\left(X_i \prec X_j\right)$ (X_i is dominated by X_j); the default behavior is a random behavior.

(2) Improved swarming behavior

The condition of carrying out the swarming behavior is changed from $if\left(Y_c/n_f > \delta Y_i\right)$ into $\mathrm{if}\left(X_c \succ X_i\right)\cap\left(n_f/N > \delta\right)$ (X_c dominates X_i, and meanwhile the position of X_c is not crowded); the default behavior is preying behavior.

(3) Improved following behavior

The condition of carrying out the following behavior is changed from $if\left(Y_{max}/n_f > \delta Y_i\right)$ into $if\left(X_{max} \succ X_i\right)\cap\left(n_f/N > \delta\right)$ (X_{max} dominates X_i, and meanwhile the position of X_{max} is not crowed), and X_{max} represents the local optimum position, which is chosen with the methods mentioned in Section 3.2.

3.2 Selection of X_{max}

MOAFSA algorithm involves the following behavior of AF, so it needs to choose the position X_{max} where there is maximum food concentration. In order to improve the distributivity and diversity of non-dominated solution set, we randomly choose a number r between 0~1. If $r < \alpha$, P (1) or P (N) will be X_{max}; otherwise, the individual with the maximum crowding distance will be chosen (where: α is a decimal number, which is adjusted according to the number N of external set NDSet. Generally, set $\alpha \approx 2/N+1$). How to choose X_{max} is as below. Where, P.dist is each individual's crowding distance and α is a small random number.

```
Xmax =Select-leader (P.dist, α )
{ r =rand;
if r < α
then Xmax={ P (1) or P (N) }
 else Xmax= P (2)
for i=3 to (N-1)
 {if P (i)>P (2)
  then Xmax= P (i);
     }end for i
  end
 }end for select
```

3.3 The Steps of MOAFSA

The detailed process of MOAFSA algorithm is as follows:

Step 1: Initialization of parameters:

Set the scale of AF swarm as POP, the visual space of AF as *Visual*, the maximum step length as *Step*, the crowed factor as δ, etc. The initial iteration time is $k \leftarrow 0$, and the upper limit of the external set is set as N.

Step 2: Position initialization of fish swarm:

The initial position X is generated randomly in the feasible region. Calculate the food concentration of AF's current position $F\left(X\right)=\left(f_1\left(X\right), f_2\left(X\right),\cdots, f_m\left(X\right)\right)$. m is the number of optimization objectives.

Step 3: Firstly, sort the value of initial AF food concentration by Function: NDSet(POP). Secondly obtain the initial non-dominated value $F\left(Xc\right)$ and the corresponding position Xc; thirdly, perform the Function: Crowding-distance-assignment ($F\left(Xc\right)$) to cut redundant individuals. Finally, store it in the external set NDSet. The initial non-dominated set is defined as $\left[FB^1{}_{ND}, XB^1{}_{ND}\right]$ (where: $FB^1{}_{ND}$ and $XB^1{}_{ND}$ are respectively the value of the initial non-dominated AF's food concentration and the corresponding position of AF). Then the initial $X^1{}_{\max}$ can be calculated by means of Function: Select-leader(POP.dist, α) introduced in Section 3.2.

Step 4: Judge whether iteration k meets the termination condition $k < K$. If not, repeat steps (5) - (7); if so, go to step (8).

Step 5: Dynamically adjust the value of *Step* according Equation (4). Then by making the k-th iteration AFs imitate swarming behaviors respectively, we get the position $X^k{}_{follow}$ and $F\left(X^k{}_{follow}\right)$, and by imitating following behaviors, we get $X^k{}_{swarm}$ and $F\left(X^k{}_{swarm}\right)$; if $X^k{}_{follow} \succ X^k{}_{swarm}$, then perform the following behavior. Otherwise, perform the swarming behavior. The default behavior is preying behavior. After this process, AF has a new position $X^k{}_{new}$ and its food concentration is $F\left(X^k{}_{new}\right)$.

Step 6: Make the construction set $F\left(X^k{}_{new}\right)$ obtained from last step execute Function: NDSet(), and then we get a new construction set $\left[FC^k{}_{ND}, XC^k{}_{ND}\right]$ in this iteration.

(7)
Merg $\left[FB^k{}_{ND}, XB^k{}_{ND}\right]$ and $\left[FC^k{}_{ND}, XC^k{}_{ND}\right]$, and then perform Function: NDSet() and Function: Crowding-distance-assignment(), thus getting a new external set $\left[FB^{k+1}{}_{ND}, XB^{k+1}{}_{ND}\right]$ in this iteration.

Step 8: Calculation termination: output the optimum solution. (i.e. the final $XB^{K+1}{}_{ND}$ and $FB^{K+1}{}_{ND}$)

4. Mathematical Simulation and Case Studies

4.1 Mathematical testing

This part uses MOAFSA to calculate the common multi-objective test functions, and compares it with classical multi-objective algorithms, such as NSGA-II and SPEA2, to test the performance and effect of the algorithm proposed. The number of times of function calculation for all algorithms is set as 25,000. The number of individuals is 100, and the maximum number of external set is 100. Where, the Cross-over rate of NSGA-II and SPEA2 algorithm is set to be 0.8, and mutation rate 0.1; MOAFSA's parameters are set in this way: $Visual = 0.3$, $Step = 0.2$, $Try\_number = 100$, $\delta = 0.618$.

4.1.1 Test functions

SCH, DEB test functions are proposed by Schaffer [1] and Deb [14], while ZDT test functions [15] are proposed by Zitzler, Deb, and Thiele in 2000. All of these test functions are classical test problems in the field of EMO. This paper adopts the 7 functions including SCH, DEB, ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 in the test.

4.1.2 Performance metrics

In order to assess the distribution and convergence of the solution set, this paper adopts the indicators of convergence consult the generational distance（GD） [16] proposed by Van Veldhuizen and Lamont, and the indicators of distributivity consults thediversity（△） proposed by Deb [3].

4.2 The mathematical Simulation

Execute 25,000 times of function calculations for each of the three algorithms above, and the condition of $e_i = 0$ is set to be whether the Euclidean distance between non-dominated solution and its nearest neighbor individual in PF_{true} is less than 0.01. Then record GD and △ obtained. All programs are run in computer with Core-2 dual-core CPU (2.1GHZ) and 2G memory. The indicator value of performance in 25,000 times of function calculations with different algorithms are shown in Table 1. Figure 2 shows us how different algorithms approach the true non-dominated solution front.

Table1. Evaluation parameters of algorithm performance

Performance matrices	Test function	Multi-objective algorithms		
		NSGA-II	SPEA-2	MOAFSA
GD	DEB	1.88E-04	2.04E-04	1.08E-04
	SCH	2.17E-03	9.75E-04	9.18E-04
	ZDT1	6.71E-03	1.67E-03	5.76E-03
	ZDT2	7.71E-03	4.72E-03	6.32E-03
	ZDT3	5.14E-03	5.13E-03	8.41E-03
	ZDT4	1.11E-01	1.56E-01	9.78E-02
	ZDT6	6.12E-02	1.87E-04	3.34E-02
△	DEB	5.37E-03	4.10E-03	3.26E-03
	SCH	6.48E-02	4.94E-03	1.58E-01
	ZDT1	9.88E-02	8.98E-02	1.18E-01
	ZDT2	1.42E-01	1.61E-01	1.21E-01
	ZDT3	1.07E-01	1.49E-01	1.53E-01
	ZDT4	5.76E-01	5.74E-01	5.60E-01
	ZDT6	2.84E-01	3.72E-03	3.82E-01

From Table 1, we can see that MOAFSA performed best in terms of the function test with low decision variables, but the algorithm's efficiency declines with the increase of decision variables of the test function.

According to convergence indicators (GD) of the three algorithms, it is clear that MOAFSA algorithm outperforms the other two algorithms in function test with low decision variable (such as SCH and DEB); the solving precision of MOAFSA is approximately the same with that of NSGA-II. All of these prove a stronger ability of MOAFSA in optimization.

But from the perspective of the distribution index (△), SPEA2 is superior to other two algorithms in some test functions; there is no big difference between the distribution indicators of MOAFSA and that of NSGA-II, but is slightly inferior to that of SPEA-II, and the distributivity of MOAFSA remains to be improved.

It can also be seen from Figure 1 that MOAFSA performs well in SCH and DEB function tests, and almost coincides with the true Pareto front; the non-dominated solution set got by MOAFSA is very close to the true Pareto frontier except in ZDT4 function test, which has a number of local extreme points. So MOAFSA is a new and effective multi-objective optimization algorithm.

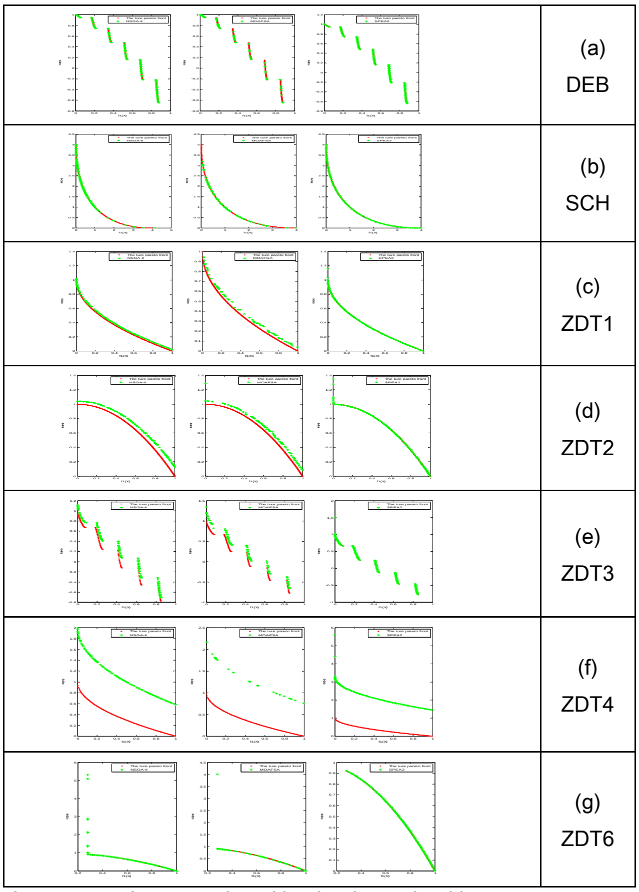

Fig.1. Pareto fronts produced by the three algorithms

4.3 Case Simulation

As for a hydropower station, the maximization energy output and its maximization of water supply volume are not compatible. The increase in water supply will inevitably result in the reduction of energy output; meanwhile, reservoir operation is a complex and nonlinear problem. Conventional algorithms cannot get the trade-off solution, so it is urgent and of great importance to apply multi-objective optimization algorithm into reservoir scheduling. This paper will take Wanjiazhai reservoir as the object of the research. It is located in the upriver of the Yellow River in China, with its main task to generate power and supply water. It also functions to control flood and prevent ice. Wanjiazhai reservoir supplies not only electricity to the power system, but also a certain amount of water to the neighboring region. Wanjiazhai reservoir is an annual storage reservoir, with a total reservoir capacity of 896 million m^3 and regulating storage volume of 445 million cubic meters. Its maximum reservoir level is 980.0 m, the normal water level is 977.0 m, the dead water level is 952 m, and the level of flood control is 966 m. Its installed capacity is 1.08 million kW, and the guaranteed output is 185,000 KW.

The data of average runoff of Wanjiazhai reservoir from 1919 to 2006 is known, and the required operating water levels for each month are as follows: it is a period to release sediment from August to September, so the water level should be between the dead water level and the flood control level; from early October to the end of July in the next year, the water level should be between the dead water level and the flood control water level. Where, storage period is from October to next April, whereas water supply period is from May to July. Reservoir provides a certain quantity of water to adjacent areas except August and September.

With operating water level meeting the aforementioned requirements, we take the water supply volume from Wanjiazhai reservoir and energy output from its hydropower station as the goals of scheduling optimization, the water levels at the beginning of each month as the initial location of the artificial fish, water supply volume and energy output as the corresponding food concentration, and the water volumes in each month as the state variable, and establish the multi-objective optimization model of reservoir scheduling; then we use MOAFSA algorithm to solve the scheduling model, which adopts the same parameters as those in Section 4.1. The target function of the reservoir scheduling model is shown below:

$$\begin{cases} E = Max \sum_{t=1}^{12} 8.3 * Q_t * H_t * \Delta t \\ W = Max \sum_{t=1}^{10} P * \Delta t \end{cases} \tag{5}$$

The constraints are:

$$\begin{cases} V_{t+1} = V_t + \left(Q_{\lambda t} - Q_t\right) \cdot \Delta t \\ Q_{t.\min} \le Q_t \le Q_{i.\max} \\ Z_{t.\min} \le Z_t \le Z_{t.\max} \\ N_{t.\min} \le AQ_t H_t \le N_{t.\max} \end{cases} \tag{6}$$

where: $Q_{\lambda t}$ and Q_t are respectively reservoir inflow and power discharge during a time period t, whose unit is m^3/s; Δt is the number of seconds in a month, whose unit is s; P is water supply flow in each month, whose unit is m^3/s; H_t is the average head during a time period t, whose unit is m; V_{t+1} and V_t are the ending and the beginning volume of reservoir storages during a time period t, whose unit is m^3; $Q_{t.\min}$ and $Q_{t.\max}$ are the minimum and maximum water discharges of reservoir during time period t, which is calculated by m^3/s; $Z_{t.\min}$ and $Z_{t.\max}$ are the minimum and maximum water levels of reservoir at the beginning of the period t, whose unit is m; $N_{t.\min}$ is the minimum power generation of reservoir during the time period t, whose unit is kW; $N_{t.\max}$ is the maximum power generation of reservoir, whose unit is kW.

The result of reservoir operation by MOAFSA is shown below in Figure 2:

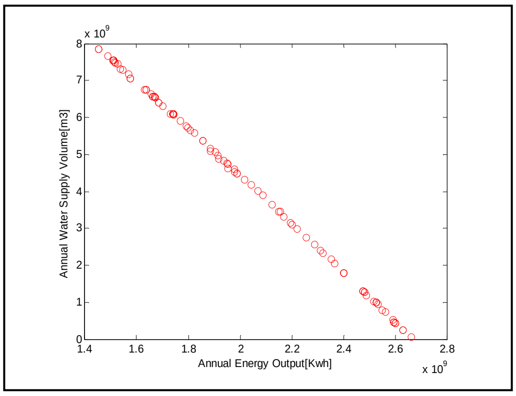

Fig.2. Non-dominated Solution Set Produced By MOAFSA

It can be seen from Figure 2 that the energy output and the water supply of Wanjiazhai reservoir are incompatible, and the increase in water supply will inevitably cause the reduction of energy output; so how to find a compromising solution that meets the requirements of all the departments is very important. We can get a set of non-dominated solutions by using MOAFSA, select some of them evenly and list it in Table 2, which can function as reference for the reservoir scheduling to make better use of the water resources. Practice has proved that MOAFSA algorithm is feasible and effective in the reservoir operation.

Table2.Non-dominated solution set of cascade hydropower stations

non-dominated solution	1	2	3	4	5
Total supply water : $10^8 m^3$	2.536	23.97	47.28	60.92	74.47
Energy output 10^8kW·h	26.32	23.11	19.52	17.41	15.25

5. Conclusion

In this paper, we propose a new multi-objective optimization algorithm—MOAFSA, which adds the non-dominated solutions solving strategies of multi-objective optimization into AFSA, uses the quick sort method to get non-dominated solution set, and cuts the external set according to the crowding distance. This paper at first introduces the idea of MOAFSA and its processing steps, and then employs MOAFSA to calculate the typical multi-objective test functions, and carries out a comparative analysis between it and the classic multi-objective optimization algorithms such as SPEA2 and NSGA-II. Results show that: (a) MOAFSA performs best in the function test with low decision variables (such as DEB and SCH), but the algorithm's efficiency declines with the increase of decision variables of the test functions; (b) the non-dominated solution set obtained through MOAFSA is very close to the true Pareto frontier except in ZDT4 function test. The performance indicators of MOAFSA are similar to or even better than those of NSGA-II; MOAFSA is proved to be an effective and new multi-objective optimization algorithm; at last, this paper applies MOAFSA in the scheduling optimization of hydropower station reservoir, and results show that it is valuable in engineering practice.

Because in MOAFSA, a considerable number of initial parameters need to be formulated, such as Step, Visual distance, Crowding factor, etc., most of which can only be got through experience or many times of experiments. Some parameters of the algorithm need to be adjusted accordingly in new function tests, thereby bringing troubles to the application of MOAFSA. So how to choose suitable initial parameters or adjust parameters adaptively deserves further study; meanwhile, this paper hasn't applied MOAFSA in the multi-objective optimization with more than two objective functions, so the performance of the algorithm needs to be further tested and inspected, which should be the focus of future study.

Acknowledgments

The authors extend their sincere gratitude for the supports from Jiangsu Province Innovation Research Fund (Fund Number: CXZZ11_0445) and the Fundamental Research Funds for the Central Universities (Fund Number: 2009B08414).

REFERENCES

[1] Schaffer, J.D., *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*, in *Proceedings of the 1st International Conference on Genetic Algorithms*. 1985, L. Erlbaum Associates Inc. p. 93-100.

[2] Srinivas, N. and K. Deb, *Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms.* Evolutionary Computation, 1994. **2**(3): p. 221-248.

[3] Deb, K., et al., *A fast and elitist multiobjective genetic algorithm: NSGA-II.* Evolutionary Computation, IEEE Transactions on, 2002. **6**(2): p. 182-197.

[4] Zitzler, E. and L. Thiele, *Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach.* Evolutionary Computation, IEEE Transactions on, 1999. **3**(4): p. 257-271.

[5] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in Proc. EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems,K.Giannakoglou,D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, Sept. 2001.

[6] Xiaolei, L., S. Zhijiang, and Q. Jixin, *An optimizing method based on autonomous animats: fish-swarm algorithm.* Systems Engineering Theory & Practice, 2002. **22**(11): p. 7.

[7] Jiang, M., et al., *Optimal Multiuser Detection with Artificial Fish Swarm Algorithm, Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, D.-S. Huang, L. Heutte, and M. Loog, Editors. 2007, Springer Berlin Heidelberg. p. 1084-1093.

[8] Jiang, M., et al., *Image Segmentation with Improved Artificial Fish Swarm Algorithm, Proceedings of the European Computing Conference.* 2009. **28**: p. 133-138.

[9] Shen, W., et al., *Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm.* Knowledge-Based Systems, 2011. **24**(3): p. 378-385.

[10] Wang, Z., *The Optimal allocation of Manufacturing Resources Capabilities in Workshop Based on AFSA*, in *Advanced Manufacturing Technology, Pts 1, 2*, J.T.J.Z.Y.J.S. Han, Editor. 2011. p. 1622-1625.

[11] Xiaolei, L., *A new intelligent optimization method-artificial fish school algorithm*. Ph.D. dissertation. 2003, Zhejiang University.

[12] Jiang, M.Y. and D.F. Yuan, *Artificial Fish Swarm Algorithm and Its Applications. Proc. of International Conference on Sensing, Computing and Automation.* 2006: p. 1782-1787.

[13] Zheng, J., et al., *A Multi-objective Genetic Algorithm Based on Quick Sort, Advances in Artificial Intelligence*, A. Tawfik and S. Goodwin, Editors. 2004, Springer Berlin / Heidelberg. p. 175-186.

[14] Deb, K., *Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems.* Evolutionary Computation, 1999. **7**(3): p. 205-230.

[15] Zitzler, E., K. Deb, and L. Thiele, *Comparison of Multiobjective Evolutionary Algorithms: Empirical Results.* Evolutionary Computation, 2000. **8**(2): p. 173-195.

[16] Van Veldhuizen, D.A., & Lamont, G. B. , *Multiobjective evolutionary algorithm research: A history and analysis.* 1998.

***Authors**: Guohua Fang, Wei Guo, Xianfeng Huang, Xinyi Si, Fei Yang, Qian Luo, Ke Yan are with the College of Water Conservancy and Hydropower Engineering, Hohai University, Nanjing, China.*
(E-mail:bounob12345@163.com;hxfhuang2005@163.com; hhufgh@126.com*; hotsauce0079@163.com; yfhhsg@yahoo.cn)*