Two-Swarm Cooperative Artificial Fish Algorithm for Bound Constrained Global Optimization*

Ana Maria A.C. Rocha,[1] M. Fernanda P. Costa,[2] and Edite M.G.P. Fernandes[1]

[1] *Algoritmi Research Centre, University of Minho, Braga, Portugal,* {arocha; emgpf}@dps.uminho.pt

[2] *Centre of Mathematics, University of Minho, Braga, Portugal* mfc@math.uminho.pt

Abstract This study presents a new two-swarm cooperative fish intelligence algorithm for solving the bound constrained global optimization problem. The master population is moved by a Lévy distribution and cooperates with the training population that follows mainly the classical fish behaviors. Some numerical experiments are reported.

Keywords: Global optimization, Swarm intelligence, Artificial fish, Lévy distribution

1. Introduction

In this study we are interested in solving the bound constrained global optimization (GO) problem using a swarm intelligence algorithm that is able to converge to the globally best point in the feasible region and requires a limited computational effort. The problem to be addressed has the form

$$\text{glob} \min_{x \in \Omega} f(x), \tag{1}$$

where f is a continuous nonlinear, possibly nonconvex function, and Ω is the hyperrectangle $\{x \in \mathrm{R}^n : l \leq x \leq u\}$. When solving complex optimization problems, like NP-hard problems, metaheuristics are able to perform rather well and generate good quality solutions in less time than the traditional optimization techniques [3]. Besides the variety of applications in some engineering areas, the motivation for the present study is the pressing and ongoing need to develop efficient algorithms for solving a sequence of problems, like (1), that emerge from a penalty function technique or an augmented Lagrangian based multiplier algorithm for constrained nonconvex global optimization, in reasonable time.

The artificial fish swarm (AFS) algorithm has been previously implemented within augmented Lagrangian paradigms [2, 10], which in turn have been compared with other metaheuristic-based penalty like algorithms to solving constrained GO problems. The numerical results have been shown that the fish swarm intelligence is a promising metaheuristic but further research is demanded so that efficiency can be improved.

*This work has been supported by FCT (Fundação para a Ciência e Tecnologia, Portugal) in the scope of the projects: PEst-OE/MAT/UI0013/2014 and PEst-OE/EEI/UI0319/2014.

2. Two-swarm cooperative paradigm

The present proposal for solving the problem (1) is a variant of the AFS algorithm. This metaheuristic relies on a swarm intelligence based paradigm to construct fish/point movements over the search space while converging to the optimal solution [2, 9, 10]. The new algorithm is termed two-swarm cooperative AFS (2S-AFS) and the crucial idea is to use two swarms (instead of just one) where each one has its own task and supplies information to the other swarm, when attempting to converge to optimality. Other multi-swarm cooperative algorithms based on a master-slave model can be found in [6, 7]. Hereafter, the terms 'point' and 'population' (of points) will be used to represent (the position of) a fish and the swarm respectively. The position of a point in the space is represented by $x_j \in \mathrm{R}^n$ (the jth point of a population) and m is the number of points in the population. The component i of a point x_j is represented by $(x_j)_i$.

2.1 Classical AFS algorithm

The initial procedure of AFS algorithm consists of randomly generating the points x_j, $j = 1, \ldots, m$ of the population, in Ω. Then, each current point x_j produces the trial point y_j according to the number of points inside its 'visual scope' (VS). This is a closed neighborhood centered at x_j with a positive radius which varies with the maximum distance between x_j and the other points. When the VS is empty, a Random Behavior is performed, and when it is crowded, one of the behaviors, Searching or Random, is performed. However, when the VS is not crowded, one of the four following behaviors is selected: Chasing, Swarming, Searching or Random. The selection depends on the objective function values of x_j when compared with the function value of the best point inside the VS, the central point inside the VS, or a randomly chosen point of the VS. To choose the population for the next iteration, the current x_j and the trial y_j are compared in terms of f. The pseudo-code for the AFS algorithm is presented below.

```
AFS algorithm
{
    randomly generate the population x_j ∈ Ω, j = 1,...,m and select x_best;
    while stopping condition is not met   {
      for each x_j, j = 1,...,m     {
        if ('visual scope' is empty)
          {compute y_j by Random Behavior}
        else if ('visual scope' is crowded)
              {compute y_j by Searching/Random Behavior}
            else
              {compute y_j by Chasing/Swarming/Searching/Random Behavior}.
        if (f(y_j) ≤ f(x_j))   {set x_j = y_j}     }
      select x_best and perform random local search around it;    }
}
```

2.2 Two-swarm cooperative AFS algorithm

In order to improve the capability of searching the space for promising regions where the global minimizers lie, this study presents a new fish swarm-based proposal that defines two-populations, each one with its task goal but always sharing information with the other: one is the master and the other is the training population. The master population aims to explore the search space more effectively, thus defining trial points from the current ones throughout a stable stochastic distribution. Depending on the number of points inside the VS of x_j of the training population, the trial point is mainly produced by the classical AFS behaviors, although in some cases – when the VS is empty and when it is crowded – the stochastic distribution borrowed from the master population is used. The overall best point is shared between both populations. The algorithm is called 2S-AFS. To be able to produce a trial y_j, from the current x_j, ideas like those of Bare-bones particle swarm optimization in [4] and the model for mutation in evolutionary programming [5], may be used:

$$(y_j)_i = \gamma + \sigma \mathrm{Y}_i \tag{2}$$

where γ represents the center of the distribution that may be given by $(x_j)_i$ or $((x_j)_i+(x_{best})_i)/2$ (the average of $(x_j)_i$ and the best point $(x_{best})_i$), σ may represent an adaptive mutation defined by the distance between $(x_j)_i$ and $(x_{best})_i$, and each Y_i is an identically distributed random variable from the Gaussian distribution with mean 0 and variance 1. We note that Y may be the random variable of another probability distribution. The standard Lévy distribution is used since it can search a wider area of the search space and generate more distinct values in the search space than the Gaussian distribution. The Lévy distribution, denoted by $\mathrm{L}_i(\alpha, \beta, \gamma, \sigma)$, is characterized by four parameters. The parameter β gives the skewness ($\beta = 0$ means that the shape is symmetric relative to the mean). The shape of the Lévy distribution can be controlled with α. For $\alpha = 2$ it is equivalent to the Gaussian distribution, whereas for $\alpha = 1$ it is equivalent to the Cauchy distribution. The distribution is stable for $\alpha = 0.5$ and $\beta = 1$. σ is the scale parameter and is used to describe the variation relative to the center of the distribution. The location parameter γ gives the center. When $\gamma = 0$ and $\sigma = 1$, we get the standard form, simply denoted by $\mathrm{L}(\alpha)$ when $\beta = 0$.

Hence, the proposal for further exploring the search space and improve efficiency is the following. The points from the master population always move according to the Lévy distribution, i.e., each trial point y_j is generated component by component $i = 1, \ldots, n$ as follows:

$$(y_j)_i = \begin{cases} (x_j)_i + (\sigma_j)_i \mathrm{L}_i(\alpha) & \text{if } rand() \leq p \\ (x_{best})_i + (\sigma_j)_i \mathrm{L}_i(\alpha) & \text{otherwise} \end{cases} \tag{3}$$

where $(\sigma_j)_i = |(x_j)_i - (x_{best})_i|$, $\mathrm{L}_i(\alpha)$ is a random number generated for each i from the standard Lévy distribution with the parameter $\alpha = 0.5$, $rand()$ is a random number generated uniformly from $[0, 1]$ and p is a user specified probability value for sampling around the best point to occur. On the other hand, each point in the training population either moves according to classical AFS behaviors if its VS is not crowded, or it moves using a Lévy distribution, as shown in (3), with $p = 0$ if the VS is empty, and $p = 1$ if the VS is crowded. Cooperation from the master population is also required if the best point belongs to the master population. The below presented algorithm is the pseudo-code for 2S-AFS algorithm.

```
2S-AFS algorithm
{
    randomly generate x_j ∈ Ω, j ∈ P ≡ {1,...,m} and select x_best;
    randomly choose x_j, j ∈ M ⊂ P, where #M = ⌊m/3⌋, and move them according to (3) with p = 0;
    while stopping condition is not met   {
       for each x_j, j = 1,...,m     {
          if (j ∈ M – point in master population)
             {compute y_j according to (3) with p = 0.5}
          else if ('visual scope' is empty)
                {compute y_j according to (3) with p = 0}
             else if ('visual scope' is crowded)
                   {compute y_j according to (3) with p = 1}
                else
                   {compute y_j by Chasing/Swarming/Searching/Random Behavior}.
          if (f(y_j) ≤ f(x_j))  {set x_j = y_j}   }
      select x_best and perform random local search around it;   }
}
```

The algorithm stops when $|f(x_{best}) - f^*| \leq 0.001$ or $NFeval > 20000$ where $f(x_{best})$ is the best solution found thus far, f^* is the known optimal solution, and $NFeval$ gives the number of function evaluations.

3. Results and Conclusions

This section aims to compare the results of the proposed 2S-AFS with those of two AFS-based algorithms on benchmark problems with acronyms BR, CB6, GP, H3, H6, SBT, S5, S7 and S10 (with n ranging from 2 to 6) [1]. The algorithm was coded in C and the results were obtained on a PC with a 2.8 GHz Core Duo Processor P9700 and 6 Gb of memory. Each problem was solved 30 times and $m = 10n$ points are used. Table 1 summarizes the results obtained in terms of the average number of function evaluations ('Nfe_{avg}') required by the algorithms to reach the optimal solution with the above defined accuracy. 'DbAFS' is a distribution-based AFS algorithm with the random local search (RLS) (see in [8]) and 'AFS' is the classical AFS with the same RLS (see also Figure 1). From the results we may conclude that 2S-AFS is quite efficient in converging to the optimal f^* on six problems but reached the maximum number of evaluations in some runs when solving problems S5, S7 and S10. These behaviors need further investigation and new strategies to enforce convergence.

	f^*	2S-AFS	DbAFS	AFS
BR	0.39789	362	690	815
CB6	-1.0316	241	293	639
GP	3.00000	494	710	830
H3	-3.86278	206	911	1273
H6	-3.32237	657	3864	6534
SBT	-186.731	415	1256	2803
S5	-10.1532	8382	1611	4568
S7	-10.4029	5793	1818	2931
S10	-10.5364	5837	1889	3067

Table 1: Comparison of AFS-based algorithms.

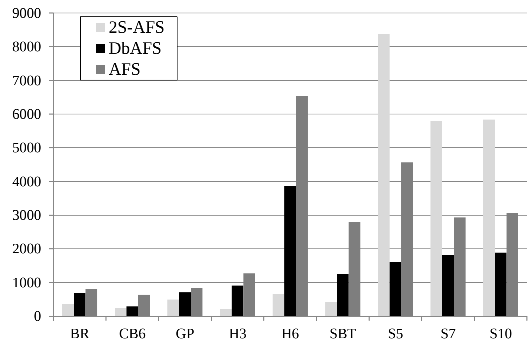

Figure 1: Bars of Nfe_{avg} for the tested algorithms.

References

[1] Ali, M.M., Khompatraporn, C., Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Glob. Optim. 31, 635–672.

[2] Costa, M.F.P., Rocha, A.M.A.C., Fernandes, E.M.G.P. (2014). An artificial fish swarm algorithm based hyperbolic augmented Lagrangian method. J. Comput. Appl. Math. 259, 868–876

[3] Gogna, A., Tayal, A. (2013). Metaheuristics: review and application. J. Exp. Theor. Artif. In. 25, 503–526

[4] Kennedy, J., Eberhart, R. (2001). Swarm Intelligence. San Mateo, CA: Morgan Kaufmann

[5] Lee, C., Yao, X. (2004). Evolutionary programming using the mutations based on the Lévy probability distribution. IEEE T. Evolut. Comput. 8, 1–13

[6] Niu, B., Zhu, Y., He, X., Wu, H. (2007). MCPSO: A multi-swarm cooperative particle swarm optimizer. Appl. Math. Comput. 85, 1050–1062

[7] K.E. Parsopoulos, K.E. (2012). Parallel cooperative micro-particle swarm optimization: A master–slave model. Appl. Soft Comput. 12, 3552–3579

[8] Rocha, A.M.A.C., Costa, M.F.P., Fernandes, E.M.G.P. (2013). Distribution based artificial fish swarm in continuous global optimization. Proc. XVI Congress APDIO, Oliveira, J.F., Vaz, C.B. (eds.) ISBN: 78-972-745-154-8, 306–312, Portugal, June 2013.

[9] Rocha, A.M.A.C., Fernandes, E.M.G.P., Martins, T.F.M.C. (2011) Novel fish swarm heuristics for bound constrained global optimization problems. Lect. Notes Comput. Sc. Vol. 6784, ICCSA 2011 Part III, B. Murgante et al. (eds.) 185–199

[10] Rocha, A.M.A.C., Martins, T.F.M.C., Fernandes, E.M.G.P. (2011). An augmented Lagrangian fish swarm based method for global optimization. J. Comput. Appl. Math. 235(16), 4611–4620