Density as the Segregation Mechanism in Fish School Search for Multimodal Optimization Problems

Salomão Sampaio Madeiro, Fernando Buarque de Lima-Neto,
Carmelo José Albanez Bastos-Filho, and Elliackin Messias do Nascimento Figueiredo

Polytechnic School of Pernambuco, University of Pernambuco, Recife, Brazil
52720-001, Recife-PE, Brazil
{ssm,fbln*,cjabf,emnf}@ecomp.upe.br

Abstract. Methods to deal with Multimodal Optimization Problems (MMOP) can be classified in three main approaches, regarding the number and the type of desired solutions. In general, methods can be applied to find: (1) only one global solution; (2) all global solutions; and (3) all local solutions of a given MMOP. The simultaneous capture of several solutions of MMOPs without parameter adjustment is still an open question in optimization problems. In this article, we discuss a density segregation mechanism for Fish School Search to enable simultaneous capture of multiple optimal solutions of MMOPs with one single parameter. The new proposal is based on vanilla version of Fish School Search (FSS) algorithm, which is inspired on actual fish school behavior. The performance of the new algorithm is evaluated and compared to the performance of other methods such as NichePSO and Glowworm Swarm Optimization (GSO) for seven well-known benchmark functions of two dimensions. According to the obtained results, presented in this article, the new approach outperforms the algorithms NichePSO and GSO for all benchmark functions.

1 Introduction

Multimodal Optimization Problems (MMOP) occur in various fields including geophysics [1], electromagnetism [2], climatology [3] and logistics [4, 5], among others. To find more than one optimal solution of MMOPs can be useful because of two main reasons [6, p. 88]: (1) to provide insights in functions landscape; and (2) to allow selections of alternative solutions, e.g. when the dynamic nature of constraints in the search space makes a previous optimum solution infeasible.

Several methods based on computational models inspired on natural processes have been proposed to deal with MMOPs. For example, Particle Swarm Optimization (PSO) [7] is an effective optimization method [8] for which several approaches have been proposed in order to make it able to capture multiple optimal solutions of MMOPs [9, 10, 11, 12, 13, 14]. Moreover, new methods to MMOPs have been proposed based on, for example, a swarm of glowworms [6].

Although several swarm based methods have been proposed to deal with MMOPs, there are still two important issues to be addressed. The first one concerns to the fact that performance of most of the proposed methods depends on manual parameter adjustment. And the second one concerns to the reduction of performance when the

Y. Tan et al. (Eds.): ICSI 2011, Part II, LNCS 6729, pp. 563–572, 2011.

dimensionality of the problem increases. In other words, many current methods, when applied to MMOPs with more than five dimensions, e.g. present low performance.

In this article, we discuss a density segregation mechanism that enables simultaneous capture of multiple optimal solutions of MMOPs put together on top of the original algorithm Fish School Search (FSS) [15]. The new proposal allow each fish to find different food sources (i.e. different optimal solutions). So, at each iteration of the algorithm, the school can be divided into subgroups. Each created subgroup corresponds to a possible solution to the multimodal problem. At the end of the execution, the set of all captured solutions is provided.

The paper is organized as follows: in Section 2, we give a detailed description of the algorithm FSS. In Section 3, we give a detailed description of the modified FSS algorithm, based on density. In Section 4, we compare the performance of the density FSS with the performance of other algorithms, anmelly, NichePSO and GSO for a set of well known benchmark of multimodal functions. The conclusion is provided and commented upon in Section 5.

2 Fish School Search

The FSS algorithm has four operators, which can be grouped in two classes: feeding and swimming. In the next subsections, we present in details the four FSS operators.

2.1 Individual Movement Operator

The individual movement operator is applied in each iteration for each fish in the school. Each fish randomly chooses a new position and evaluates it by using the fitness function. The fish will only move to that calculated position if it is more advantageous than the current one; otherwise, it stays at the same position. The next candidate position is determined as shown in (1):

$$x_i(t+1) = x_i(t) + rand(-1,1)\, step_{ind}(t), \tag{1}$$

where x_i(t) is the current position of the fish in dimension i, x_i(t+1) is the new calculated position of the fish for dimension i, and $rand()$ is a function that return numbers uniformly distributed in a given interval. The $step_{ind}$ is calculated as a percentage of x_{max}) for all dimension i. $step_{ind}$ decreases linearly during iterations by using $step_{ind}(t+1) = step_{ind}(t) - (step_{ind\,initial} - step_{ind\,final})/iterations$ in order to improve the exploitation ability in later iterations, where $iterations$ is the number of iterations used in the simulation. The $step_{ind\,initial}$ and $step_{ind\,final}$ are the initial and the final individual movement step, respectively. Note that the $step_{ind\,initial}$ must be higher then $step_{ind\,final}$ in order for the gradual shift from exploration to exploitation modes of operation along the iterations.

2.2 The Feeding Operator

The weight is a metaphor to quantify the success of the search process for individual fish. The heavier a fish is, the higher is the probability for it to be in a good region of the search space (i.e. aquarium). The amount of food that a fish eats depends on the improvement in its fitness and the largest improvement in the fitness of the entire school. The weight of a fish is updated according to $W_i(t+1) = W_i(t) + \Delta f_i / max(\Delta f)$, where $W_i(t)$ is the weight of the fish i, (Δf_i) is the difference of the fitness at current and new position for the fish i, $max(\Delta f)$ is a function that returns the maximum difference of the fitness values among all the fish. One should remember that $\Delta f_i = 0$ for a fish that does not perform the individual movement at the current iteration.

2.3 Collective Instinctive Movement Operator

Only fish that successfully performed individual movements, i.e. $\Delta x_i \neq 0$, influence the resulting direction of the school movement. The resulting direction (I) is evaluated by using (2). After that, all fish of the school must update their positions according to (3). The collective instinctive movement operator, at each iteration of the algorithm, tends to guide the whole school in the direction of movement taken by the fish that found the largest portion of food in its individual movement (i.e. to regions of the space search in which it was discovered the large amounts of food).

$$I(t) = \frac{\sum_{i=1}^{N} \Delta x_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}, \tag{2}$$

$$x_i(t+1) = x_i(t) + I(t). \tag{3}$$

2.4 Collective Volitive Movement Operator

The collective volitive movement operator is important to: (1) balance the trade-off between exploration and exploitation; and (2) avoid the algorithm to be trapped in local optima. The school contraction is applied as a step - inward drift - to every fish position with regard to the school barycenter. Conversely, the school dilatation is applied as a step outwards. The fish-school barycenter is obtained by considering all fish positions and their weights, as shown in (4). All fish must update their positions according to (5) when the total weight of the school increases at the current iteration. On the other hand, if the total weight of the school remains constant or reduces at the current iteration, all fish must update their positions according to (6).

$$B(t) = \frac{\sum_{i=1}^{N} x_i W_i(t)}{\sum_{i=1}^{N} W_i(t)}, \tag{4}$$

$$x(t+1) = x(t) - step_{vol}\, rand(0,1) \frac{(x(t) - B(t))}{distance(x(t), B(t))}, \tag{5}$$

$$x(t+1) = x(t) + step_{vol}\, rand(0,1) \frac{(x(t) - B(t))}{distance(x(t), B(t))}, \tag{6}$$

where $distance()$ is a function which returns the Euclidean distance between the barycenter and the fish current position, $step_{vol}$ is a predetermined step used to control the displacement from/to the barycenter.

The $step_{vol}$ must be in the same order of magnitude of the step used in the individual movement. As $step_{vol}$ is multiplied by a factor drawn from the uniform distribution in interval [0,1] with expected value equal to 0.5, which is usually twice the $step_{ind}$ value.

3 Density as the Segregation Mechanism in FSS

The modifications applied to FSS are presented as follows.

3.1 Feeding Operator

In the new proposal, as opposed to what occurs in vanilla FSS, Δf_i is shared among all fish. The objective of that sharing process is to locate areas for which fish cooperate (i.e. share food) and then, gather together all cooperating fish in the same subgroup. The sharing mechanism of Δf_i among other fish j is performed according to (7). It depends on two factors: (1) the normalized distance $d_{R_{ij}} = \frac{d_{ij}}{\min d_{ik}}, k \neq i$, where $k \in 1,2,\ldots,N$ (i.e. $d_{R_{ij}} \geq 1$), and (2) the number q_{ij} of fish k for which $d_{ik} < d_{ij}$ (i.e. the density of fish around fish i), including fish i.

$$\Delta f_i = P_i \sum_{j=1}^{N} \frac{1}{(d_{R_{ij}})^{q_{ij}}} = \frac{\Delta f_i}{\sum_{j=1}^{N} \frac{1}{(d_{R_{ij}})^{q_{ij}}}} = P_i, \tag{7}$$

In (7), P_i must be evaluated for each fish i and it represents the amount of food the fish i will receive after sharing Δf_i. The other fish j will receive $\frac{1}{(d_{R_{ij}})^{q_{ij}}}$ of P_i, as given in (8). In (7), when $i = j$, we have $d_{R_{ij}} = 0$ and $q_{ij} = 0$, resulting in 0^0. For this case, computationally, we consider $0^0 = 1$. Note that if, for a given fish i, we have $\min d_{ik} = 0$, we decided to consider $\min d_{ik} = 4,9E^{-324}$ as this is the lowest possible value for the numerical precision using the data type double in our computer set up.

$$C(i,j) = \frac{P_i}{(d_{R_{ij}})^{q_{ij}}} = \frac{\Delta f_i}{(d_{R_{ij}})^{q_{ij}} \sum_{k=1}^{N} \frac{1}{(d_{R_{ik}})^{q_{ik}}}}. \tag{8}$$

According to (7) and (8), each fish j will receive an amount of food exponentially smaller according to $(d_{R_{ij}})^{-q_{ij}}$. This equation is based on the one used by Martinetz and Schulten [16, p. 520] in the step (iv) of their proposed algorithm for creating Topology Representing Networks (TRN). For density FSS, the expression $e^{\frac{-k_i}{\lambda}}$ of [16, p. 520] was adapted to $(d_{R_{ij}})^{-q_{ij}}$, in order to quantity the amount of food a fish j will receive because the successful foraging behavior of another "colleague" fish i. The greater the value of q_{ij} (i.e. the greater the density of fish around fish i), the smaller will be the amount of food avaiable for fish j. That means that crowded areas exert little influence over other fish.

At the end of each iteration, each fish i received an amount of food given as $C(j,i)$ from other fish j that successful found food. The sum of each amount $C(j,i)$ for all other fish j corresponds to the total amount of food fish i received in a given iteration. Then, the weight $W_i(t)$ of fish i at the t^{th} iteration is updated according to (9).

$$W_i(t+1) = W_i(t) + \sum_{j=1}^{Q} \frac{\Delta f_j}{(d_{R_{ij}})^{q_{ij}} \sum_{k=1}^{N} \frac{1}{(d_{R_{jk}})^{q_{jk}}}}, \tag{9}$$

where Q is the quantity of fish that successful found food at the t^{th} iteration. In this new proposal, differently from a real fish in nature, we assumed that the weight of all artificial fish do not decrease along iterations.

3.2 Memory Operator

In the algorithm derived here, each fish i has a memory $M_i = \{M_{i1}, M_{i2}, \ldots, M_{iN}\}$, where N is the number of fish in the school, and M_{ij} quantifies the influence of one fish j over the fish i. M_{ij} depends on the total amount of food the fish i received because of the foraging behavior of fish j (i.e. $C(j,i)$) along the entire execution of the algorithm. The bigger is $C(j,i)$, the greater will be the influence of fish j over fish i. This exerted influence manifests itself in terms of how synchronized will be the behavior of a fish i regarding the foraging behavior of another fish j.

After the Feeding Operator is computed, the Memory Operator updates M_{ij} as shown in (10). In (10), $0 \leq \rho \leq 1$ is a parameter that controls the influence of one fish over every other. For example, if the value of ρ is close to 1, in general, just after a relatively small number of iterations (e.g. 10 iterations), the memory of each fish may be greatly changed. That is, fish here learn and forget rather quickly.

$$M_{ij}(t+1) = (1-\rho)M_{ij}(t) + \frac{\Delta f_j}{(d_{R_{ij}})^{q_{ij}} \sum_{k=1}^{N} \frac{1}{(d_{R_{jk}})^{q_{jk}}}} = (1-\rho)M_{ij}(t) + C(j,i). \tag{10}$$

3.3 Collective Instinctive Movement Operator

For density based FSS, the resultant behavior I_i for each fish i is evaluated as shown in (11). In (11), I_i is a sum of the directions taken by each fish j during the Individual Movement Operator weighted by M_{ij}. Note that, contrary of FSS, even though a fish does not locate food (*i.e.* $\Delta x_j = 0$), it would influence the resultant behavior I_i of other fish i. In other words, even if fish j does not locate food, fish i will mimic its behavior (*i.e.* remain stationary) according to the memorized value M_{ij}.

$$I_i(t) = \frac{\sum_{j=1}^{N} \Delta x_j M_{ij}}{\sum_{k=1}^{N} M_{ik}}. \tag{11}$$

3.4 Operator for Partitioning the Main Fish School

In our new approach, at each iteration of the algorithm, the main school is partitioned into subgroups. One fish i will be in the same subgroup of other fish j if and only if:

$$M_{ij} = \max_{k=1,2,\ldots,N} M_{ik} \; \vee \; M_{ji} = \max_{k=1,2,\ldots,N} M_{jk}, \tag{12}$$

where N is the number of fish in the main school. Therefore, fish i is into the same subgroup of fish j if and only if fish j is the fish that exerts the largest influence over fish i or fish i is the fish that exerts the largest influence over fish j.

The algorithm for the partition of the main school into subgroups is illustrated in Algorithm 1. Through this procedure, a fish i is chosen randomly in the main school. After that, all other fish j of the main school that are into the same subgroup of fish i, according to the definition in (12), are removed from the main school and put in the same subgroup of fish i. Then, for each fish j, all other fish k that are into the same subgroup of fish j are removed from the main school. This process of selection of fish that are into the same subgroup of fish i is repeated until all the fish into the subgroup of fish i have been removed from the main school. Then, another fish i is chosen randomly from the main school and the procedure is repeated until all the fish have been removed from the main school.

Algorithm 1. Pseudo code of Partition Operator

```
while There is fish in the main school do
  Choose a fish i randomly in the main school
  Create a new subgroup S_i
  Put fish i in subgroup S_i
  Remove fish i from the main school
  Find other fish j in the main school that satisfies (12)
  while there exists fish j in the main school do
    Put fish j in subgroup S_i
    Remove fish j from the main school
    Set i = j
    Find other fish j in the main school that satisfies (12)
  end while
end while
```

3.5 Individual Movement Operator

In density FSS, the new update mechanism to update the length of the step of the fish is given in (13).

$$step_{ind_i}(t+1) = decay_i * step_{ind_i}(t), \tag{13}$$

$$decay_i = decay_{min} - \left(\frac{R_i(t) - min(R_j(t))}{max(R_j(t)) - min(R_j(t))}\right)(decay_{min} - decay_{max}(t)), \tag{14}$$

$$decay_{max}(t) = decay_{max_{ini}} \left(\frac{decay_{max_{end}}}{decay_{max_{ini}}}\right)^{\frac{t}{T_{max}}}, \tag{15}$$

$$R_i(t) = \sum_{j=1}^{Q} \frac{\Delta f_j}{(d_{R_{ij}})^{q_{ij}} \sum_{k=1}^{N} \frac{1}{(d_{R_{jk}})^{q_{jk}}}}, \tag{16}$$

where $step_{ind_i}(0) = step_{init}$, $0 \leq decay_{min} \leq 1$, $0 \leq decay_{max_{ini}} \leq 1$, and $0 \leq decay_{max_{end}} \leq 1$ are parameters of the algorithm and $decay_{max_{end}} < decay_{max_{ini}} <$

$decay_{min}$. $decay_{min}$ is the decay to be applied to the step of the fish that received the smallest amount of food in a subgroup. $decay_{max}$ decays exponentially as given in (15) and it is used to reduce the step size of the fish that received the largest amount of food in a subgroup. For the other fish in a given subgroup, the value of $decay_i$, where $decay_{max} < decay_i < decay_{min}$, is evaluated as given in (14).

3.6 Collective Volitive Movement Operator

In density based FSS, the Collective Volitive Movement Operator is performed independently for each created subgroup. The barycentre is calculated for each subgroup based on the weight of fish as given in (4). Then, aiming to allow a progressive convergence of each subgroup around a potential solution of one MMOP, each fish "swims" in the direction of the barycentre of its subgroup according to (17). To avoid premature convergence of subgroups around regions for which there is no solution, the size of the step to be performed for a fish in the barycentre direction of its subgroup varies according to the value of $decay_{max}(t)$.

$$x(t+1) = x(t) + (1 - decay_{max}(t))(B(t) - x(t)). \tag{17}$$

4 Experiments

In this section the performance of our new approach is compared to the performance of NichePSO [10] and GSO [6] for benchmark functions with two and more dimensions and with a finite number of optimal solutions.

4.1 Methodology

For the experiments in this section, we used the set of multimodal functions shown in Table 1.

For all benchmark functions, we consider that: (1) the entities are disposed uniformly on the search space; to ensure a uniform distribution of the entities in the search space, we used Faure sequences [18] to generate a uniform sequence of pseudo-random numbers; (2) the number of iterations for density FSS is halved regarding the number of iterations of the algorithms NichePSO and GSO for all experiments, since in density FSS each fish performs two calls to the objective function at each iteration; (3) if the normalized distance between two captured optima i and j, for all optima, is less than 0.01, we assume those optima as being the same optimal solution; in this case, only the fittest optimum will be taken, and the other one will be discarded. The normalized distance $d_N(i,j)$ between two points i and j is given as $d_N(i,j) = \sqrt{\frac{(x_N{}^i - x_N{}^j)\cdot(x_N{}^i - x_N{}^j)}{D}}$, where $x_N = (\frac{x_1}{x_{1_{max}}}, \frac{x_2}{x_{2_{max}}}, \ldots, \frac{x_1}{x_{D_{max}}})$, D is the number of dimensions of one MMOP, and $x_{k_{max}}$ is the upper bound of the dimension k; (4) for all selected optima, we considered that density FSS and NichePSO have captured an optimal solution k if the normalized distance between k and the optima closest to k is less than 0.005; in GSO [6, p. 99], an optimal solution k is captured when at least three glowworms are located at a distance less than $\varepsilon = 0.05$ from k. In this paper, the value of ε was modified to 0.005, following the procedure used for the algorithms NichePSO and density FSS, as described earlier.

Table 1. Multimodal benchmark functions. The domain of the search space and the corresponding number of peaks for that domain are given in the second and the third columns, respectively.

Function	Domain	No. of peaks	Source
$F_1(X) = \sum_{i=1}^{m} cos^2(X(i)), X \in \mathbb{R}^m$	$[-\pi, \pi]^m$	3^m	[6]
$F_2(x,y) = cos^2(x) + sin^2(y)$	$[-5,5]^2$	12	[11, 6]
$F_3(x) = 1 + \sum_{i=1}^{n} \frac{x_i^2}{4000} - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-29,29]^2$	124	[17]
$F_4(x,y) = 200 - (x^2+y^2-11)^2 - (x+y^2-7)^2$	$[-6,6]^2$	4	[10, 6]
$F_5(x,y) = 3(1-x)^2 e^{-[x^2+(y+1)^2]} - 10(\frac{x}{5} - x^3 - y^5) e^{-[x^2+y^2]} - (\frac{1}{3}) e^{-[(x+1)^2+y^2]}$	$[-3,3]^2$	3	[6]
$F_6(x,y) = \sum_{i=1}^{Q} a_i e^{-b_i((x-x_i)^2+(y-y_i)^2)}$, where $Q = 10$, $a_i = 1+2\vartheta$, $b_i = 2+\vartheta$, $x_i = -5+10\vartheta$, $y_i = -5+10\vartheta$, and ϑ is a random value uniformly distributed in [0,1]	$[-5,5]^2$	10	[6]
$F_7(X) = 10n + \sum_{i=1}^{m} \left[x_i^2 - 10\cos(2\pi x_i)\right], X \in \mathbb{R}^m$	$[-5,5]^2$, $[-3,3]^m$	100, 6^m	[6]

The parameter configuration for NichePSO was the same used in [14, p. 2300]. In [14, p. 2300], $c_1 = c_2 = 1.2$, w linearly decreases from 0.7 to 0.2, $\delta = 10^{-4}$, $\mu = 10^{-2}$ and $\varepsilon = 0.1$. Those values are used for all experiments performed in this paper. For GSO, we used the same configuration as described in [6, p. 99]: $\rho = 0.4$, $\gamma = 0.6$, $\beta = 0.08$, $n_t = 5$, $s = 0.03$, $l_0 = 5$. The value of the parameter $r_s = 2$ was chosen for all experiments based on the results presented for GSO in [6, pp. 109–110]. For density FSS, the parameter values were: $\rho = 0.3$, $step_{init} = 0.05$, $decay_{min} = 0.999$, $decay_{max_{ini}} = 0.99$, and $decay_{max_{end}} = 0.95$. All those values were determined based on tedious numerical experiments performed earlier.

4.2 Performance Comparison among NichePSO, GSO and Density FSS

In this section, for each function, we vary both the number t of iterations of the algorithms and the number n of entities in the swarm. In the experiments in this section, we set t = 50, 100, 150, ..., 500. We choose $t_{max} = 500$ iterations because, for some numerical experiments, the performance of the algorithms do not improve significantly after 500 iterations. For the functions F_1, F_2, F_4, F_5 and F_6, we used $n =$ 5, 10, 15, ..., 200, 210, 220, ..., 350. Therefore, for those functions, there are 550 (10*55) combinations between t and n. For the functions F_3 and F_7, we used $n =$ 5, 10, 25, 50, 100, 150, ..., 1300, 1400 and $n =$ 5, 10, 25, 50, 100, 150, ..., 1000, respectively. Then, for F_3, there are 300 (10*30) combinations and for the F_9 there are 230 (10*23) combinations. For each combination, we collect the mean value and the standard deviation of the number of optimal solutions captured by the algorithms in 30 trials.

Table 2. Comparison of the performance of the algorithms NichePSO, GSO and density FSS regarding the percentage of the total number of combinations for which the algorithms captured on average more than 95% of the number of optimal solutions of one MMOP

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
NichePSO	68.18%	34.55%	0.667%	80.09%	82.73%	0.00%	0.00%
GSO	72.75%	52.00%	9.33%	73.09%	78.18%	0.00%	0.00%
dFSS	74.00%	64.55%	34,67%	86.91%	83.45%	0.00%	23.48%

In order to compare the performance of the algorithms NichePSO, GSO and density FSS we choose the metric: percentage of the total number of combinations for which the algorithms captured on average more than 95% of the number of optimal solutions (i.e. number of peaks) of one MMOP. Table 2 summarizes the performance of the algorithms. In general, as one can note in Table 2, density FSS outperformed the algorithms NichePSO and GSO for all functions used in this Section, regarding the first metric. For the function F_7, for example, density FSS captured on average more than 95 optimal solutions for 23.48% of the total number of combinations, whereas NichePSO and GSO failed to capture more than 95 optimal solutions for all combinations.

5 Conclusions

In this paper, the algorithm FSS was adapted to locate simultaneously multiple optima of MMOP. We were able to produce a new mechanism (and algorithm), based on the principle of density, that affords the segregation for splitting the fish school and, consquently, is able to locate multiple optima. Two new operators are proposed for the partition of the fish school into subswarms, such that each created subswarm corresponds to one potential optimal solution of a given MMOP. At the end of the execution of the algorithm, a set of captured optima of a given MMOP is clearly produced.

The experimental results demonstrate that FSS based on density is a far better approach to MMOP than NichePSO and GSO. The reason for that is the evident ability of density FSS to simultaneously capture multiple optima without heavy parameterization additional costs. The highlights of the current proposal are: (1) it outperforms NichePSO and GSO for all benchmark functions; (2) it has the ability to tackle MMOPs of more than two dimensions with the need of manual parameter adjustments.

Acknowledgments

The authors acknowledge the financial support from CAPES, CNPq and University of Pernambuco for scholarships, support and travel grants.

References

[1] Koper, K., Wysession, M., Wiens, D.: Multimodal function optimization with a niching genetic algorithm: A seismological example. Bulletin of the Seismological Society of America 89(4), 978–988 (1999)

[2] Dilettoso, E., Salerno, N.: A self-adaptive niching genetic algorithm for multimodal optimization of electromagnetic devices. IEEE Transactions on Magnetics 42(4), 1203–1206 (2006)

[3] El Imrani, A., Zine El Abidine, H., Limouri, M., Essaid, A.: Multimodal optimization of thermal histories. Comptes Rendus de l'Academie de Sciences - Serie IIa: Sciences de la Terre et des Planetes 329(8), 573–577 (1999)

[4] Luh, G.-C., Chueh, C.-H.: Job shop scheduling optimization using multi-modal immune algorithm. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 1127–1137. Springer, Heidelberg (2007)

[5] Naraharisetti, P., Karimi, I., Srinivasan, R.: Supply chain redesigns - multimodal optimization using a hybrid evolutionary algorithm. Industrial and Engineering Chemistry Research 48(24), 11094–11107 (2009)

[6] Krishnanand, K., Ghose, D.: Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intelligence 3(2), 87–124 (2009)

[7] Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. Micro Machine and Human Science, 39–43 (1995)

[8] Shi, Y., Eberhart, R.C.: An empirical study of particle swarm optimization. In: IEEE Congress on Evolutionary Computation, pp. 1945–1960 (1999)

[9] Parsopoulos, K., Vrahatis, M.N.: Modification of the particle swarm optimizer for locating all the global minima. In: Karny (ed.) Artificial Neural Networks and Genetic Algorithms, pp. 324–327 (2001)

[10] Brits, R., Engelbrecht, A.P., van den Bergh, F.: A niching particle swarm optimizer. In: Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning, pp. 692–696 (2002)

[11] Parsopoulos, K., Vrahatis, M.N.: On the computation of all global minimizers through particle swarm optimization. IEEE Transactions on Evolutionary Computation 8(3), 211–224 (2004)

[12] Brits, R., Engelbrecht, A., van den Bergh, F.: Locating multiple optima using particle swarm optimization. Applied Mathematics and Computation 189(2), 1859–1883 (2007)

[13] Ozcan, E., Yilmaz, M.: Particle swarms for multimodal optimization. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 366–375. Springer, Heidelberg (2007)

[14] Engelbrecht, A., Van Loggerenberg, L.: Enhancing the nichepso. In: IEEE Congress on Evolutionary Computation, CEC, pp. 2297–2302 (2007)

[15] Bastos-Filho, C., de Lima Neto, F., Lins, A., Nascimento, A., Lima, M.: Fish school search. In: Chiong, R. (ed.) Nature-Inspired Algorithms for Optimisation. SCI, vol. 193, pp. 261–277. Springer, Heidelberg (2009)

[16] Martinetz, T.M., Schulten, K.J.: Topology representing networks. Neural Networks 7(3), 507–522 (1994)

[17] Griewank, A.: Generalized descent for global optimization. Journal of Optimization Theory and Applications 34, 11–39 (1981)

[18] Thiemard, E.: Economic generation of low-discrepancy sequences with a b-ary gray code. Technical report, Departement de Mathematiques, Ecole Polytechnique Federale de Lausanne, CH-1015 Lausanne, Switzerland (1998)