A new Heuristic of Fish School Segregation for Multi-Solution Optimization of Multimodal Problems

Marcelo Gomes Pereira de Lacerda, Fernando Buarque de Lima Neto
Polytechnic School of Pernambuco
University of Pernambuco
Recife, Brazil
mgpl@ecomp.poli.br, fbln@ecomp.poli.br

***Abstract*—This work presents a new heuristic for fish school segregation applied to the Fish School Search algorithm (FSS), aiming to serve as a basis for the creation of a new multi-solution optimization method for multimodal problems. In this new approach, the weight of the fish is used as a population segregation element, allowing the heaviest fishes, i.e the most successful ones, to swim,i.e. to move in the search space, more independently and the lightest ones to be guided by the heaviest ones. The obtained results showed that this new approach is able to find a good amount of solutions in the search space, overcoming the three techniques used for comparison in 6 of 7 benchmark functions. Moreover, it can be seen that the new approach requires less computational effort to obtain excellent results. Another advantage of the new approach is that there is no need for an addition of operators in the original FSS. Even though this new version of multimodal FSS does not have an ideal coverage, which causes the return of many "extra" solutions, the sole use of the weight of the fishes, i.e. a readily available information, as a population segregation operator is an economical and good alternative to be considered upon multi-solution problems. This specially taking into account the expediency of the method and that the detected candidate solutions are mostly false-positives, which can be more easily pruned than the addition of false-negatives.**

Keywords—Heuristic Search; Multi-Solution Optimization; Multimodal Problems; Fish School Search.

I. INTRODUCTION

Optimization tasks are present in many situations where information technology is required. Managers, for example, must take decisions aiming to maximize the company's income, but there are multiple ways for that to be achieved. Racing car teams use their resources in a way which the car endures the least damaged possible, and this can be pursued by various different ways. These are just simple real life examples where multi-model optimization tasks are required. Formally, optimization is defined as a system adjustment aiming to obtain the best possible output. However, in many cases, a "good" result is fair enough [1], not alone, the only possible.

Optimization problems can be classified in many different ways: according to the number of variables and their types, according to the linearity level of the objective function, the dynamicity of this function, the existence or not of constraints, the number of objective function and, finally, the number of optimal solutions. According to the number of optimal solutions, optimization problems can be classified as unimodal or multimodal. Unimodal problems are characterized as problems with only one global optimal solution. Multimodal problems are, thus, characterized by the existence of more than one global optimal solution, the objective function. Considering only minimization cases, multimodal optimization problems can be defined as the search process for all the local optimum x_L^* (solutions) in the function f as defined by (1), where L is the closest region to x_L^* and $\mathbb{R}^n$ is an n-dimensional real numbered search space [1].

$$f(x_L^*) \leq f(x), \forall x \in L, L \subset \mathbb{R}^n \quad (1)$$

Multimodal problems occur in many different fields, such as geophysics, electromagnetism, climatology and logistics [2].

Optimization algorithms are computational techniques that search for solutions for a certain problem, this represented by an objective function. Several nature inspired methods have been developed in order to tackle the multitude of available problems. A successful set of these techniques are known as population based algorithms (PBA), due to their characteristics of using a group of artificial entities to collectively and in a coordinated way perform the search.

Swarm Intelligence (SI), the most prominent of PBA for searching, can be viewed as a system in which the interaction between the very simple particles of the population generates complex functional patterns [3]. Some of the best known algorithms within SI are: Particle Swarm Optimization (PSO) [4], Ant Colony Optimization (ACO) [5], Artificial Bee Colony (ABC) [6], Bacterial Foraging Algorithm (BFA) [7] and Fish School Search (FSS) [8].

Fish School Search (FSS), which was proposed by Bastos et al. [8], is, in its basic version, an unimodal optimization algorithm inspired on the collective behavior of fish schools. The mechanisms of feeding and coordinated movement, which provides protection to every fish within the swarm against external predators, were used as inspiration to create the collective search mechanism. The main idea is to make the fishes (individuals) to swim toward the direction of the positive gradient in order to gain weight. Collectively, the heavier fishes are more influent in the search process as a whole as the barycenter of the school gradually moves towards better places in the search space.

Niching algorithms are a special member of the SI family that were developed in order to solve multimodal problems, finding multiple solutions at the same time. Some of the

 ISBN: 978-1-61208-269-1

existent niching techniques are the CPSO (Craziness PSO) [9], Multi_PSOer (Multi_Optimizer PSO) [10], NichePSO [11], GSO (Glowworm Swarm Optimization), the latter an SI algorithm based on the social behavior of the glowworms [12].

Density based Fish School Search (dFSS) is another niching algorithm based on the FSS proposed by Madeiro et al. [2]. It tackles multimodal problems elegantly using the concept of food sharing, rather than single fish foraging. The nearby fishes influence and group up around each other along the search process, which leads to the splitting of the whole swarm into sub-swarms. This makes possible the sought detection of multiple solutions. The major problem of that approach is the addition of two new operators onto the original FSS and, consequently, a substantial additional computational cost to the search process.

The current paper proposes another heuristic for the original FSS as an attempt to create a new multimodal version of the algorithm, trying to overcome the necessity of additional operators. In this new approach, the weight of the fishes themselves determine the relations between fishes and consequent strength of the mutual influence among each other. The rationale is simple yet creative: the higher is the difference between the weights of two fishes, the higher is the probability of existing a relation between them and the stronger is the influence from the heavier onto the lighter. This devised mechanism leads to school segregation fulfilling the scope of multiple optimization tasks.

In Section II and Section III, FSS and dFSS Algorithms are detailed, respectively. Section IV introduced the new segregation method based on the weight of fishes. Section V details the experiments whose results are presented in Section VI. And in Section VII conclusions and future works are presented.

II. FISH SCHOOL SEARCH

Fish School Search is inspired by the collective behavior of natural fish schools. In nature, many species live in groups, in order to increase their chances to survive to external threats and also to forage more effectively. In fish schools, the individuals work collectively as a single organism but do possess some local freedom. This combination accounts for fine as well as greater granularities during their search for food.

In FSS, the success of the search process is represented by the weight of each fish. In other words, the heavier is an individual, the better is its represented solution. The weight of the fish is updated throughout the feeding process. A second means to encode success in FSS is the radius of the school. The mechanism to update this is explained in the following subsections together with the other operators. But in advance it is noteworthy to mention that by contracting or expanding the radius of the school FSS can automatically switches between exploitation and exploration, respectively. The pseudo-code of FSS is provided in Fig. 1. Other details such as stopping conditions and minor variations to increase performance are left out as they are not important here.

```
P: Fish population;
while Stopping condition is not met do:
    for each fish in P do:
        Run the individual movement;
    for each fish in P do:
        Run the feeding process;
    for each fish in P do:
        Run the collective instinctive movement;
    Calculate the fish school's barycenter;
    for each fish in P do:
        Run the collective volitive movement;
Return the best solution found;
```

Figure 1. Pseudo-code of FSS.

A. Individual Movement Operator

In the Individual Movement Operator, each fish moves randomly and independently, but toward the positive gradient. In other words, this operator is executed only if the new position is better than the previous one, with regards to the objective function. This movement is described by (2), where $x_{ij}(t+1)$ is the new value of the dimension j in the position vector of the individual i, $x_{ij}(t)$ is the old value, r is a random value between 0 and 1 and $step_{ind}(t)$ is the step size on time t. The new step size is calculated through (3), where $step_{ind_{init}}$ and $step_{ind_{final}}$ are the initial and final step sizes and $iterations$ is the maximum number of iterations.

$$x_{ij}(t+1) = x_{ij}(t) + r\, step_{ind}(t), \quad (2)$$

$$step_{ind}(t+1) = step_{ind}(t) - \frac{step_{ind_{init}} - step_{ind_{final}}}{iterations}. \quad (3)$$

B. Feeding Operator

As mentioned before, the feeding operator is responsible for the weight update of all fishes, which is made accordingly with the *fitness* increase obtained after the individual movement. This update process is defined by (4), where Δf_i, is the fitness variation after the Individual Movement of the fish i, and $\max(\Delta f)$ is the maximum fitness variation in the whole population.

$$W_i(t+1) = W_i(t) + \frac{\Delta f_i}{\max(\Delta f)}. \quad (4)$$

C. Collective Instinctive Movement Operator

The Collective Instinctive Movement Operator is the first collective movement in the algorithm. A single vector is computed as a vectorial average of all individual movements and added to the position vector of all fish. This operator is defined by (5), where N is the population size, Δx_k and $\Delta f(x_k)$ is the position variation and the fitness variation of the individual of index in the Individual Movement.

 ISBN: 978-1-61208-269-1

$$x_{ij}(t+1) = x_{ij}(t) + \left(\frac{\sum_{k=1}^{N} \Delta x_{kj} \Delta f(x_k)}{\sum_{k=1}^{N} \Delta f(x_k)}\right). \quad (5)$$

D. Collective Volitive Movement Operator

In this step, the population must contract or expand, using as reference the barycenter of the fish school, which is calculated according to (6), where $W_i(t)$ is the weight of the fish i on time t. The total weight of the whole population must be calculated in order to decide if the fish school will contract or expand. If the total weight increased after the last Individual Movement, the school as a whole will contract in order to execute a finer search, as it means that the search process has been successful. Otherwise, the population will expand, meaning that the search process is not qualitatively improving. This could be due to a bad region of the search space or the school is trapped in a local minimum (hence, it should try to scape from it). The contraction and expansion processes are defined by (7) and (8), respectively, where

$$B_j(t) = \frac{\sum_{i=1}^{N} x_{ij} W_i(t)}{\sum_{i=1}^{N} W_i(t)}, \quad (6)$$

$$x_{ij}(t+1) = x_{ij}(t) - step_{vol} rand(0,1) \frac{\left(x_{ij}(t) - B_j(t)\right)}{distance\left(x_i(t), B_j(t)\right)}, \quad (7)$$

$$x_{ij}(t+1) = x_{ij}(t) - step_{vol} rand(0,1) \frac{\left(x_{ij}(t) - B_j(t)\right)}{distance\left(x_i(t), B_j(t)\right)}. \quad (8)$$

III. DENSITY BASED FISH SCHOOL SEARCH

The dFSS is an FSS based multi-solution optimization algorithm for multimodal problems. In it, the food found by any fish during the search process is shared among ”peers”. Moreover, each fish stores the amount of food given by others. Using this information, each fish decides to which ones it will be linked in order to establish sub-populations. In dFSS the operator resemble FSS, but the collective movements are made in a way that the sub-populations explore in parallel many different niches [2]. The pseudo-code of dFSS is provided in Fig. 2.

Two operators were added to the ones existing in the original FSS in order to create the multimodal version. They are: the Memory Operator and the Operator for Partitioning the Main Fish School. Some adjustments were made to all the already existent operators of the FSS to account for the various sub-swarm (and not only one anymore). The operators are described next.

A. Individual Movement

The Individual Movement mechanism remains almost the same, but individual step size update method was changed. It is calculated by (9), (10), (11) and (12), where $step_{ind_i}(t)$ is the individual step of the fish i, $decay_{min}$ is the minimum decay rate, $decay_{max_{final}}$ and $decay_{max_{ini}}$ are the final and initial max decay rate, respectively, T_{max} is the maximum number of iterations, q_{ij} is the number of fishes that lie between the fishes i and j, including the fish i and $d_{R_{jk}}$ is the relative distance between the fishes j and k. In other words, $d_{R_{jk}} = \frac{d_{jk}}{\min(d_{ij})}$, where j represents any other fish different from i.

$$step_{ind_i}(t+1) = decay_i(t) * step_{ind_i}(t), \quad (9)$$

$$decay_i = decay_{min} - \left(\frac{R_i(t) - \min(R_j(t))}{\max(R_j(t)) - \min(R_j(t))}\right)(decay_{min} - decay_{max}(t)), \quad (10)$$

$$decay_{max}(t) = decay_{max_{ini}} \left(\frac{decay_{max_{final}}}{decay_{max_{ini}}}\right)^{\frac{t}{Tmax}}, \quad (11)$$

$$R_i(t) = \sum_{j=1}^{Q} \frac{\Delta f_i}{\left(d_{R_{ij}}\right)^{q_{ij}} \sum_{k=1}^{N} \frac{1}{\left(d_{R_{jk}}\right)^{q_{jk}}}}. \quad (12)$$

The fishes are initialized;
The fitnesses of the fishes are calculated;
The distances between the fishes are calculated;
***while** Stopping criteria is not met:*
  for each** Fish in the population **do:
    Run the Individual Movement;
    ***if** The fish's fitness increased **then**:*
      for each** Fish in the population **do:
        Run the feeding operator;
        Update the fish's weight;
        Run the memory operator;
  for each** Fish in the population **do:
    Run the Collective Instinctive Movement;
    Determine the most influent fish for the given fish;
  Run the division of the main population process;
  for each** Generated sub-population **do:
    Calculate the barycenter;
  for each** Fish in the population **do:
    Update the Individual Movement step size;
    Run the Volitive Collective Movement of the fish;
  Calculate the fitness for each fish in the population;
  Calculate the distances between the fishes;
  Update the value of the variable $decay_{max}(t)$;
Returns the best individual of each sub-population formed in the last iteration as the solutions located by the algorithm;

Figure 2. Pseudo-code of dFSS.

 ISBN: 978-1-61208-269-1

B. Feeding Operator

In the dFSS, the fishes share their food. The amount of food shared by the fish i to the fish j is defined by (13), which is very similar to (12).

$$C(i,j) = \frac{\Delta f_i}{\left(d_{R_{ij}}\right)^{q_{ij}} \sum_{k=1}^{N} \frac{1}{\left(d_{R_{jk}}\right)^{q_{jk}}}}. \quad (13)$$

C. Memory Operator (new to original FSS)

The memory for all fishes has the role of storing with which ones they have shared more food in the past. For fish i, the bigger is the amount of food M_{ij} shared by the fish j, the bigger is the influence of j upon i. The memory of an individual i is represented by a vector $M_i = \{M_{i1}, M_{i2}, \ldots, M_{iN}\}$, where N is the number of fishes in the population. In (14), the food sharing process is defined, where $\rho \in [0,1]$.

$$M_{ij}(t+1) = (1-\rho)M_{ij}(t) + C(j,i). \quad (14)$$

D. Collective Instinctive Movement

In the Collective Instinctive Movement of the original FSS, all the fishes move to the same direction. In the dFSS, each fish calculates its own direction according to the most influential fishes within its own sub-swarm (the fishes connected to it), as described by (15).

$$x_{ij}(t+1) = x_{ij}(t) + \left(\frac{\sum_{k=1}^{N} \Delta x_{kj} M_{ik}(t)}{\sum_{k=1}^{N} M_{ik}(t)}\right). \quad (15)$$

E. Operator for Partitioning the School (new to original FSS)

This operator is responsible for the partitioning of the main fish school into smaller ones. A fish i is part of the same fish school of the fish j if and only if i is the most influent fish for j or j is the most influent one for i. Moreover, if i is part of the same fish school of j, the contrary is also true.

Before starting with the operator, there must be a reference for each individual in the fish school, because this operator includes removing operations. The process of definition of sub-populations begins with the random choice of a fish i from the fish school. The selected fish must be removed. Then, the fishes which will be in the same fish school as the fish i must be defined and a link between them and i must be created. After this, these fishes must be also removed. The same process must happen for the fishes that had been just removed: the fishes that must be in the same fish school of a given removed fish must be removed as well and a link between them must be created. When there is no fish that can be in the same fish school as a given fish, a new fish must be randomly chosen among all the fishes that is still in the main fish school, so that the process can start again. This procedure must be executed until the main fish school is empty.

F. Collective Volitive Movement

This operator has the role to drive the produced fish schools to the found niches. Differently from the original FSS, in the dFSS, the volitive movement occurs inside each fish school. This movement is defined by (16), where $decay_{max}(t)$ is defined by (11) and $B_{kj}(t)$ is the barycenter of the fish school of which the fish i is part.

$$x_{ij}(t+1) = x_{ij}(t) + (1 - decay_{max}(t))\left(B_{kj}(t) - x_{ij}(t)\right). \quad (16)$$

IV. Using the Weight of the Fish as a Segregation Element

In this paper, we propose a new approach for a multi-solution optimization version of multimodal problems for the FSS algorithm as an alternative for the dFSS. A new operator is included in the original algorithm, in addition to some changes in the already existent operators that were needed.

The main flow of this new approach is the same as in the FSS, except for the addition of the new operator right before the Individual Movement. This new operator is responsible for defining the links between the fishes, which defines the "guide-guided" relationship. This relationship defines a dependency relationship between the fishes which is directly proportional to the difference between the weights of the fishes. The bigger is this difference, the bigger is the dependency from the lighter to the heavier one. From these relationships, a desirable and fair independent behavior of the heaviest fishes in the main fish school emerges. The heavier is a fish relatively to the others, the more independent it is. The lighter is the fish, the more dependent it is as it is understandable, given that its light weight suggests lack of success.

In the next sections, the new approach is described. It is important to point out that the individual movement and the feeding operator have not suffered any changes, hence no explanation for them are needed (again).

A. "Guide-Guided" Relationship Links Definition Operator

As mentioned before, this operator has the role of defining links between fishes. These links define "guide-guided" relationships. Fishes which are linked to each other are called *partners*. In this relationship, the lighter fish is guided by the heavier one.

In (17), $W_i(t)$ and $W_r(t)$ are the weight of the fishes i and r, respectively and C_r and C_i are the number of fishes already linked to the fishes r and i, respectively, right before the probability calculation.

$$p_{ir}(t) = \frac{W_i(t)}{W_r(t) C_r C_i}. \quad (17)$$

 ISBN: 978-1-61208-269-1

S: Set of fishes from the main fish school;
T: Temporary set of fishes from the main fish school;
Removes every old links between the fishes;
while** S is not empty **do:
Choose randomly a fish i in S and remove it from this set;
Restart T with all the fishes from the population;
while** T is not empty **do:
Choose randomly a fish r from T and remove the individual from this set;
if** i is different from r **then:
if** $p_{ir}(t)$, which is calculated through (17), is bigger or equal to a random value generated in the interval [0,1] **and** there is not any already created link between r and i **then:
i becomes partner of r, where i guides r;

Figure 3. Pseudo-code of the new guide-guided operator.

The variables C_r *and* C_i were included to avoid the monopolization of too-heavy fishes. The semantics for that is: the greater is the number of connections of a fish, the lower is the probability of creating a new link. This mechanism was created in order to make the search process execute wider searches in the given search space.

Considering a fix indexation of the array of individuals in the main fish school, if the choice of fishes for link formation was made sequentially, the fishes with lower index would have a higher probability of being chosen, because, in the beginning, there is almost no links between the give fish and the other ones. Moreover, the first fishes in the queue to choose partners would have more chances create links, since there is almost no links between the fishes in the main fish school. In order to obtain a fair link formation process between fishes, the process of choosing fishes to find new partners is made randomly, as occurs with the choice of fishes to decide if the link between the given fish and the chosen fish must be created. In this way, any dependency of the indexation of the fishes within the array of individuals is completely avoided.

B. Collective Instinctive Movement

In this operator, the algorithm inherited the basic rationale from the same operator in the FSS. The difference in the new approach is that a given fish must calculate its own movement vector based only on its guiders and itself. The operator is defined by (18), where k is the index of the guiders of i and Ng_i is the number of guiders of i.

$$x_{ij}(t+1) = x_{ij}(t) + \left(\frac{\Delta x_{ij} \Delta f(x_i) + \sum_{k=1}^{Ng_i} \Delta x_{kj} \Delta f(x_k)}{\Delta f(x_i) + \sum_{k=1}^{Ng_i} \Delta f(x_k)} \right). \quad (18)$$

C. Collective Volitive Movement

The Collective Volitive Movement also inherited the basics from the FSS algorithm. However, each fish must calculate its own reference point to work as the barycenter of the original FSS. These personal reference points are calculated using solely the partners of each fish. The barycenter calculation is described by (19), where k is the index of the partners of i and $Npart_i$ is the number of partners of i. The mechanism of expansion and contraction is the same as the FSS.

$$B_i(t) = \frac{x_{ij} W_i(t) + \sum_{k=1}^{Npart_i} x_{kj} W_k(t)}{W_i(t) + \sum_{k=1}^{Npart_i} W_k(t)}. \quad (19)$$

IV. EXPERIMENTS DESCRIPTION

Three multi-solution optimization algorithms were used for comparison purposes: NichePSO, GSO and dFSS. We will refer to the approach proposed in this paper as thisFSS. The parameters setup is described in Table I.

Four metrics were used to evaluate and compare the performance of thisFSS with the other algorithms: (i) the number of solutions found by the algorithm; (ii) configuration with the lowest computational cost able to find 95% of the correct solutions or more; (iii) amount of solutions returned by the algorithm; and (iv) amount of solutions returned by the algorithm that is not part of the objective function's solution set. However, in the last two metrics, the algorithm GSO was not used for comparison.

Seven objective functions were used for benchmarking. These functions are detailed in [2], including their domains and number of peaks (correct solutions). Notice that here they are numbered differently: Equal Peaks A as F1; Equal Peaks B as F2; Griewank as F3; Himmelblau as F4; Peaks as F5; Random peaks as F6; Rastrigin as F7. It is important to highlight that these functions were not taken from real-world scenarios

The performance of the algorithms was analyzed varying the size of the population and the number of iterations. The values used for the population size are defined depending on the objective function. They are defined on Table II. The values for the number of iterations are the same for all the functions: 50,100,150,..,500. However, for the dFSS and thisFSS, just the half of each value is used: 25,50,75,..,250. This is due to the double call of the fitness function per iteration made by the FSS, what was inherited by both versions. This way, it is possible to make fair comparisons. 30 simulations for each algorithm (for each objective function) were made for each possible combination between population size and number of iterations. The average value of each metric for each combination is calculated.

It is important to define what is considered as a returned solution. For thisFSS, the best individual of a fish school is one solution. However, in the end of the last iteration, a different method for defining links between fishes is used. According to this method, two fishes must be linked if the

 ISBN: 978-1-61208-269-1

distance between them is lower than 0.01. This value was defined empirically.

TABLE I. PARAMETERS SETUP

Algorithm	Parameters						
NichePSO	C1	C2	ω		δ	μ	ε
	1.2	1.2	Linear decreasing from 0.7 to 0.2.		10^{-4}	10^{-2}	0.1
GSO	ρ	γ	β	n_t	s		l_0
	0.4	0,6	0.08	5	0.03		5
dFSS	ρ	$step_{init}$	$decay_{min}$	$decay_{max_{init}}$	$decay_{max_{final}}$		
	0.3	0,05	0.999	0.99	0.95		
thisFSS	$step_{ind}$		$step_{vol}$				
	Linear decreasing from 0.4 to 0.		Linear decreasing from 0.025 to 0.				

TABLE II. VALUES FOR THE POPULATION SIZE

Functions	Values
Equal Peaks A, Equal Peaks B, Himmelblau, Peaks and Random Peaks.	5, 10, 15, .., 200, 210, 220, .., 350
Griewank	5, 10, 25, 50, 100, 150, 200, 250, .., 1300, 1400
Rastrigin	5, 10, 25, 50, 100, 150, 200, 250, .., 1000

Among the returned solutions, it must be defined which ones correspond to the correct solutions of the objective function. One solution $\vec{k}$ was successfully found if the normalized distance between the solution returned by the function and $\vec{k}$ is lower than 0.005. Moreover, a solution returned by the algorithm of which distance to the closest solution of the objective function is higher than 0.01 is considered as a wrong solution. (20) and (21) explains the normalized distance calculation between the points $\vec{i}$ and $\vec{j}$, where D is the number of dimensions of the objective function.

$$d_N(\vec{i},\vec{j}) = \sqrt{\frac{\left(x_N^i - x_N^j\right)\cdot\left(x_N^i - x_N^j\right)}{D}}, \quad (20)$$

$$x_N = \left(\frac{x_1}{x_{1_{max}}}, \frac{x_2}{x_{2_{max}}}, \ldots, \frac{x_D}{x_{D_{max}}}\right). \quad (21)$$

V. RESULTS AND DISCUSSION

In Table III, the superiority of thisFSS in comparison with the other algorithms can be noticed in five out of seven objective functions. In Random Peaks, none of the methods were able to find 95% of the correct solutions or more in none of the combinations between the values for population size and the values for the number of iterations. For this function, the comparison results can be analyzed in Table IV. As it can be noticed, thisFSS overcame all other algorithms in every interval defined in the first column.

In Table V, the configuration of the lowest computational cost which is able to find 95% of the correct solutions or more for each algorithm in each objective function is shown. Contents of each cell in this table are represented as x/y(z); 'x' is the number of calls to the fitness function; 'y' is the population size and 'z'=$\sqrt{x^2+y^2}$. In other words, 'z' is the Euclidian distance between a given configuration represented as a bi-dimensional vector [x,y] and the theoretical configuration with the lowest cost possible, [0,0]. Obviously, a configuration with 0 calls to the fitness function and 0 individuals is not feasible. According to this table, thisFSS was able to find 95% of the correct solutions or more with less computational effort in five out of the six functions used for comparison.

Regarding to the absolute value of the difference between the number of solutions returned by the algorithm and the number of solutions of the objective functions (Table VI and Table VII), thisFSS obtained lower values than the NichePSO in six out of seven functions, but in all the functions obtained a higher value than the dFSS algorithm. Moreover, regarding to the percentage of wrong solutions relative to the total amount of solutions returned by the algorithms, thisFSS overcame the NichePSO in five out of seven functions and lost for the dFSS in all the functions.

TABLE III. PERCENTAGE OF CONFIGURATIONS THAT ARE ABLE TO FIND 95% OF THE CORRECT SOLUTIONS.

Objective Function	NichePSO	GSO	dFSS	thisFSS
Equal Peaks A	68.18%	72.55%	74%	**82.18%**
Equal Peaks B	34.55%	52%	64.55%	**82.18%**
Griewank	0.67%	9.33%	34.67%	**50%**
Himmelblau	80.09%	73.09%	86.91%	**88.36%**
Peaks	82.73%	78.18%	**83.45%**	82.54%
Random Peaks	0%	0%	0%	0%
Rastrigin	0%	0%	23.48%	**56.52%**

TABLE IV. PERCENTAGE OF CONFIGURATIONS THAT ARE ABLE TO FIND THE NUMBER OF SOLUTIONS OF THE FUNCTION RANDOM PEAKS DEFINED IN THE FIRST COLUMN.

Number of Solutions	NichePSO	GSO	dFSS	thisFSS
>=7.6	0%	0.18%	0.18%	**3.09%**
>=6.65 and <7.6	7.6	31.82%	16.36%	**38.55%**
>=5.7 and <6.65	6.65	42%	32.91%	**38.82%**
Total>=5.7	73.82%	49.45%	77.55%	**91.63%**

TABLE V. CONFIGURATION WITH THE LOWEST COST ABLE TO FIND 95% OF THE CORRECT SOLUTION OF THE OBJECTIVE FUNCTIONS OR MORE.

Objective Function	NichePSO	GSO	dFSS	thisFSS
Equal Peaks A	100/95 (137.93)	100/80 (128.06)	100/130 (164.01)	**100/40 (107.70)**
Equal Peaks B	100/210 (232.59)	150/125 (195.25)	150/110 (186.01)	**100/50 (111.80)**
Griewank	100/1400 (1403.57)	150/1150 (1202.08)	400/600 (721.11)	**150/550 (570.09)**
Himmelblau	100/60 (116.62)	100/110 (148.66)	100/75 (125)	**100/45 (109.66)**
Peaks	**50/70 (86.02)**	100/80 (128.06)	100/90 (134.53)	100/25 (103.08)
Random Peaks	-	-	-	-
Rastrigin	-	-	400/500 (640.31)	**150/400 (427.2)**

These results show that the thisFSS is able to find more correct solutions than the other ones in almost all the objective

functions used as benchmark, but is not able to satisfactorily attract the individuals of the population to the peaks that represent these solutions without decreasing the number of correct solutions found. This leads to a big amount of different returned solutions and a high percentage of false-positive solutions. Even though the results of thisFSS are much better than the results of the NichePSO and GSO; comparing to dFSS using the last two metrics, thisFSS did not obtain good results, although in the first two metrics, the new approach overcame the first one in six out of seven functions.Some attempts to speed up the convergence of the fish schools in the peaks of the search space without losing the number of correct solutions were made, such as the addition of memory for the links of each fish, instead of removing every link in the beginning of each iteration, but the results did not improve as expected. Moreover, different probability functions for creation of links between fishes were tested, but, still, the results did not improve and, in some cases, worsened. So, the results showed in this paper were the best ones found so far.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the sole use of FSS fish weight as the segregation factor for producing a new multimodal version of the original algorithm. The obtained results showed that the new approach is able to find a bigger amount of correct solutions in most of the functions, when compared to the NichePSO, GSO and even dFSS. However, the technique also returns false-positives which is a disadvantage if compared to the dFSS. However, the proposed version here does require only one economical operator on top of FSS (and not two as in dFSS). This is a highlight, as computing resources can be a problem in high dimensional problems .

TABLE VI. ABSOLUTE VALUE OF THE DIFFERENCE BETWEEN THE NUMBER OF SOLUTIONS FOUND BY THE ALGORITHM AND THE NUMBER OF SOLUTIONS OF THE OBJECTIVE FUNCTIONS.

Objective Function	NichePSO	dFSS	thisFSS
Equal Peaks A	45.73	**0.29**	16.5
Equal Peaks B	42.42	**0.31**	28.3
Griewank	109.38	**28.82**	138.3
Himmelblau	54.52	**0.59**	37
Peaks	59.05	**3.31**	29.2
Random Peaks	51.64	**1.87**	25.7
Rastrigin	70.66	**25.74**	53.5

TABLE VII. PERCENTAGE OF WRONG SOLUTIONS RELATIVE TO THE TOTAL AMOUNT OF SOLUTIONS RETURNED BY THE ALGORITHMS IN EACH OBJECTIVE FUNCTION (FALSE-POSITIVES).

Objective Function	NichePSO	dFSS	thisFSS
Equal Peaks A	86.27%	**4.94%**	27.77%
Equal Peaks B	82.71%	**13.65%**	48.39%
Griewank	2.68%	**0.81%**	36.29%
Himmelblau	98.51%	**16.34%**	77.56%
Peaks	97.50%	**55.47%**	87.81%
Random Peaks	95.07%	**19%**	65.83%
Rastrigin	3.86%	**1.09%**	24.30%

From all presented results, it is possible to conclude that the use of the weight of the fishes is indeed a good alternative to the concept of density used in the dFSS, but one needs to consider the trade-offs in both ways. Another advantage of the algorithm proposed here when compared to the dFSS is that in the latter, a shared memory was added to all fishes in order to make it possible the definition of relations between them. This is no longer necessary here because the weights are already existent in all fishes of the original FSS. And they are used to define these links, what makes the new approach simpler than the first one. Moreover, the calculation of food sharing in dFSS uses topological information, such as the calculation of Euclidian distances, what makes it highly sensitive to dimensional explosion , i.e. high dimensional problems. And this does not happen at all in the new approach since the weight is independent of the number of dimensions of the objective function.

As future work, different mechanisms will be tested in order to reduce the number of returned false-positives. The new approach will also be tested for a higher number of dimensions. Moreover, a computational cost evaluation will be performed.

REFERENCES

[1] A. P. Engelbrecht, , Computational Intelligence, An Introduction, 1st ed., Wiley & Sons, 2007.

[2] S. S. Madeiro, F. B. Lima Neto, C.J.A.B Filho , and E. M. N Figueiredo, , "Density as the segregation mechanism in fish school search for multimodal optimization problems", Advances in Swarm Intelligence. ICSI 2011. International Conference on, vol. 2, 2011, pp. 563-572.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm Intelligence: from natural to artificial systems. Oxford University Press, Inc., 1999.

[4] J. Kennedy. and, R. Eberhart., "A new optimizer using particle swarm theory", Micro Machine and Human Science, International Symposium on, 1995, pp. 39-43.

[5] M. Dorigo. Optimization, learning and natural algorithms. PhD Thesis – Politecnico di Milano, 1992.

[6] D. Karaboga, and B. Basturk "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm.", Journal of Global Optimi-zation, 2006, pp. 459–471.

[7] K. M. Passino., "Biomimicry of bacterial foraging for distributed optimization and control.", IEEE Control Systems Magazine, v. 22, n. 3, 2002, pp. 52–67.

[8] C. J. A. B Filho., F. B. de Lima Neto, A. J. C. C.. Lins, A. I. S. Nascimento., and M. P. Lima, , "A novel search algorithm based on fish school behavior," Systems, Man and Cybernetics, SMC 2008. IEEE International Conference on, 2008, pp. 2646-2651.

[9] E. Ozcan,. and, M. Yilmaz,, "Particle swarms for multimodal optimization." Lecture Notes in Computer Science, v. 4431, 2007, pp. 366–375.

[10] L. Li, L. Hong-Qi, and X. Shao-Long,, "Particle swarm multi_optimizer for locating all local solutions.", Evolutionary Computation. IEEE Congress on, 2008, pp. 1040–1046.

[11] R. Brits., A. P. Engelbrecht, and F. van der Bergh,, "Locating multiple op-tima using particle swarm optimization.", Applied Mathematics and Computation, v. 2, n. 189, 2007, pp. 1859–1883..

[12] K. Krishnanand., and D. Ghose,, "Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions.", Swarm Intelligence, v. 3, n. 2, 2009, pp. 87–124.

 ISBN: 978-1-61208-269-1