Optimizing multi-plateau functions with FSS-SAR (Stagnation Avoidance Routine)

João Batista Monteiro Filho,
Isabela M. C. de Albuquerque,
Fernando Buarque de Lima Neto and
Filipe V. S. Ferreira
Department of Computer Engineering
Polytechnical School of Pernambuco
Recife, Brazil
Email: {jbmf,imca,fbln,fvsf}@ecomp.poli.br

***Abstract*—In this article we investigate the effectiveness of a stagnation avoidance routine devised for Fish School Search algorithm, which is a novel nature inspired population based search procedure effective for continuous optimization problems. Here we used and modified either Vanilla and a niching version of the algorithm in order to improve their exploratory ability with the introduction of a stochastic worsening allowance behavior within the local search operator, originating the stagnation avoidance routine. Two sets of multi-plateau objective functions were defined in order to evaluate the performance improvement in multi-plateau search spaces, a common and challenging threat for Combinatorial Optimization algorithms. The main goal was to improve the convergence capability of the algorithm when solving very smooth or plateau containing search spaces. Instances of the multi-plateau functions as well as a set of benchmark test problems were solved. Results show that the proposed modification for Fish School Search is quite effective in abbreviating as well as improving the convergence rate of the algorithm.**

1. Introduction

A sizeable number of optimization problems from Industry and Academia are of Combinatorial Optimization nature [1]. Many of these problems are known to be NP-Hard. Hence, different meta-heuristic procedures have been studied in order to solve them in practical time [2]. Some of the mentioned problems commonly tackled by researchers include the travelling salesman problem [3], [4], the knapsack problem [5] and the job shop schedulling problem [6], [7], among many others.

Combinatorial Optimization Problems consist in finding an optimal solution from a finite search space [1], which means that, in such type of problem, the search space is composed by a finite number of different plateaus.

Fish School Search (FSS) algorithm, presented originally in 2008 in the work of Bastos-Filho and Lima-Neto et al. [8], is a population based continuous optimization technique inspired in the behavior of fish schools while looking for food. Each fish in the school represents a solution for a given optimization problem and the algorithm utilizes information of every fish to guide the search process to promising regions in the search space as well as avoiding early convergence in local optima.

Ever since the original version of FSS algorithm was developed, many modifications were performed in order to tackle different issues such as multi-objective optimization [9], multi-solution optimization [10] and binary search [11]. Among those, a novel niching and multi-solution version known as wFSS was recently proposed [12].

As it will be further detailed in the following sections, FSS intrinsically uses accumulation of success during the search instead of local information like other meta-heuristic techniques such as PSO [13]. It means that the algorithm depends on steady improvement of some fishes in the population in order for them to better guide the search process. This fact can become a disadvantage in the cases of very smooth or plateau containing search spaces, which is common when solving combinatorial problems utilizing continuous search procedures, as can be seen in the work of Hamta et al. [14].

To the best of the authors knowledge, the aforementioned issue regarding efficiency of FSS when trying to optimize smooth search spaces is still open. Hence, this paper is intended to propose a seminal modification in the original FSS intended to allow the algorithm to converge even when trying to solve plateau containing objective functions. This modification was also applied in wFSS and then a comparison between original and proposed versions and Particle Swarm Optimization (PSO) algorithm was performed in multi-plateau objective functions as well as in well known continuous optimization test problems.

This paper is organized as follows: section 2 provides an overview of Fish School Search algorithm and its niching version, wFSS. Section 3 introduces the proposed modifications in order to increase the exploration ability of the technique, the Stagnation Avoidance Routine (SAR). Section 4 describes the multi-plateau objective functions utilized in order to evaluate the convergence capacity of the algorithms tested. Section 5 presents the tests performed and the results achieved.

2. Fish School Search Algorithm: Vanilla and w versions

2.1. FSS Vanilla

FSS is a population based search algorithm inspired in the behavior of swimming fishes that expand and contract while looking for food. Each fish n-dimensional location represents a possible solution for the optimization problem. The algorithm makes use of weights for all fishes which represent cumulative account on how successful has been the search for each fish in the school.

FSS is composed of feeding and movement operators, the latter being divided into three sub-components, which are:

1) **Individual component of the movement:** Every fish in the school performs a local search looking for promising regions in the search space. It is done as represented by the following equation:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \textbf{rand(-1,1)}\, step_{ind}, \quad (1)$$

where $\mathbf{x}_i(t)$ and $\mathbf{x}_i(t+1)$ represent the position of fish i before and after the individual movement operator, respectively. **rand(-1,1)** is an uniformly distributed random numbers array with the same dimension as $\mathbf{x}_i(t)$ and values varying from -1 up to 1. $step_{ind}$ is a parameter that defines the maximum displacement for this movement. The new position $\mathbf{x}_i(t+1)$ is only accepted if the fitness of fish i improves with the position change. If it is not the case, $\mathbf{x}_i(t)$ remains the same and $\mathbf{x}_i(t+1) = \mathbf{x}_i(t)$.

2) **Collective-instinctive component of the movement:** An average of displacements performed within individual movements is calculated based on the following:

$$\mathbf{I} = \frac{\sum_{i=1}^{N} \Delta\mathbf{x}_i \Delta f_i}{\sum_{i=1}^{N} \Delta f_i}. \quad (2)$$

Vector $\mathbf{I}$ represents the weighted average of the displacements of each fish. It means that fishes which experienced a higher improvement will attract other fishes into its current position.

After vector $\mathbf{I}$ computation, every fish will be encouraged to move according to:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{I}. \quad (3)$$

3) **Collective-volitive component of the movement:** This operator is used in order to regulate exploration/exploitation abilities of the school during the search process. First of all, barycenter $\mathbf{B}$ is calculated based on the position $\mathbf{x}_i$ and weight W_i of each fish:

$$\mathbf{B}(t) = \frac{\sum_{i=1}^{N} \mathbf{x}_i(t) W_i(t)}{\sum_{i=1}^{N} W_i(t)}, \quad (4)$$

and then, if total weight given by the sum of weights of all N fishes in the school $\sum_{i=1}^{N} W_i$ has increased from last to current iteration, the fishes are attracted to the barycenter according to equation 5. If the total school weight has not improved, fishes are spread away from the barycenter according to equation 6:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) - step_{vol}\textbf{rand(0,1)} * \frac{\mathbf{x}_i(t) - \mathbf{B}(t)}{distance(\mathbf{x}_i(t), \mathbf{B}(t))}, \quad (5)$$

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + step_{vol}\textbf{rand(0,1)} * \frac{\mathbf{x}_i(t) - \mathbf{B}(t)}{distance(\mathbf{x}_i(t), \mathbf{B}(t))}, \quad (6)$$

where $step_{vol}$ defines the maximum displacement performed with the use of this operator. $distance(\mathbf{x}_i(t), \mathbf{B}(t))$ is the euclidean distance between fish i position and the school barycenter. **rand(0,1)** is an uniformly distributed random numbers array with the same dimension as $\mathbf{B}$ and values varying from 0 up to 1.

Besides movement operators, it was also defined a feeding operator used in order to update the weights of every fish according to:

$$W_i(t+1) = W_i(t) + \frac{\Delta f_i}{max(|\Delta f_i|)}, \quad (7)$$

where $W_i(t)$ is the weight parameter for fish i, Δf_i is the fitness variation between the last and new positions and $max(|\Delta f_i|)$ represents the maximum absolute value of fitness variation among all fishes in the school.

W is only allowed to vary from 1 up to W_{scale}, which is a user defined attribute. Weights of all fishes are initialized with the value $W_{scale}/2$.

The pseudo-code for FSS is the following:

Initialize user parameters
Initialize fishes positions randomly
while Stopping condition is not met **do**
    Calculate fitness for each fish
    Run individual operator movement
    Calculate fitness for each fish
    Run feeding operator
    Run collective-instinctive movement operator
    Run collective-volitive movement operator
end while

The parameters $step_{ind}$ and $step_{vol}$ decay linearly according to:

$$step_{ind}(t+1) = step_{ind}(t) - \frac{step_{ind}(initial)}{It_{max}}, \quad (8)$$

and similarly:

$$step_{vol}(t+1) = step_{vol}(t) - \frac{step_{vol}(initial)}{It_{max}}, \quad (9)$$

where $step_{ind}(initial)$ and $step_{vol}(initial)$ are user defined initial values for $step_{ind}$ and $step_{vol}$, respectively. It_{max} is the maximum number of iterations allowed in the search process.

2.2. wFSS

Introduced in the work of Lima-Neto and Lacerda [12], wFSS is a weight based niching version of FSS intended to produce multiple solutions. The niching strategy is based on a new operator called Link Formator used to define leaders for the fishes in order to form sub-schools.

Link Formator operator works according to the following: each fish a chooses randomly another fish b in the school. If b is heavier than a, then a now has a link with b and follows b (i.e. b leads a). Otherwise, nothing happens. However, if a already has a leader c and the sum of the weights of the followers of a is higher than the weight of b, then a stops following c and starts following b. In each iteration, if a becomes heavier than its leader, the link will be broken.

The pseudo-code for wFSS is:

```
Initialize user parameters
Initialize fishes positions randomly
while Stopping condition is not met do
    Run link formation operator
    Calculate fitness for each fish
    Run individual operator movement
    Calculate fitness for each fish
    Run feeding operator
    Run collective-instinctive movement operator
    Run collective-volitive movement operator
end while
```

Besides Link Formator operator inclusion, some modifications were performed in the movement operators. Collective-instinctive component of the movement becomes:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \rho\mathbf{I}, \tag{10}$$

with:

$$\mathbf{I} = \frac{(\Delta\mathbf{x}_i\Delta f_i + L\Delta\mathbf{x}_l\Delta f_l)}{\Delta f_i + L\Delta f_l}, \tag{11}$$

where L is 1 if the fish i has a leader and 0 otherwise. $\Delta\mathbf{x}_l$ and Δf_l are the displacement and fitness variation of the leader of fish i. Further:

$$\rho = \frac{currentiteration}{It_{max}}. \tag{12}$$

Collective-volitive component of the movement is also modified in a sense that the barycenter is now calculated for each fish with relation to its leader. If the fish does not have a leader, its barycenter will be its actual position. The aforementioned is shown in equation 13:

$$\mathbf{B}(t) = \frac{\mathbf{x}_i(t)W_i(t) + L\mathbf{x}_l(t)W_l(t)}{W_i(t) + LW_l(t)}. \tag{13}$$

3. Stagnation Avoidance Routine

As mentioned before, a modification was proposed in original FSS in order to make it to improve its exploration ability. In the original version of the algorithm, Individual movement component is only allowed to move a fish when its fitness improves. However, in a very smooth search space, there would be many moving trials with no success and the algorithm could fail to converge.

Further, Collective-volitive movement was designed to regulate exploration/exploitation ability of the algorithm along the search process, however, in order to do so, this behavior depends on the possibility of the school total weight to reduce. If it does not happen, only equation 5 will be utilized in this operator. This means that the ability of attracting fishes to school barycenter in order to exploit the search space will always predominate with relation to the ability of spreading the school away from the barycenter in order to allow exploration.

To solve these issues, we introduced a parameter α for which $0 \leq \alpha \leq 1$ in the Individual component of the movement. α decays exponentially along with the iterations and measures a probability for a worsening allowance for each fish. It means that, every time a fish tries to move to a position that does not improve its fitness, a random number is chosen and if it is smaller than α the movement is allowed. However, only fishes which presented improvement in their fitnesses within the Individual component of the movement can contribute to $\mathbf{I}$ calculation utilized within Collective instinctive movement. In this case, $\mathbf{I}$ will be calculated according to:

$$\mathbf{I} = \frac{\sum_{i \in N} \Delta\mathbf{x}_i\Delta f_i}{\sum_{i \in N} \Delta f_i}, \tag{14}$$

where N is the set of all fishes which improved their fitnesses in the last Individual movement performed.

This modification is intended to improve the algorithm exploration ability by allowing stochastic worsening movements. However, as parameter α decays exponentially along the iterations, this effect is intense only in the beginning of the search process and becomes irrelevant after some iterations.

4. L-Shape and C-Shape Step-Plateau functions

Two sets of functions containing plateaus were defined in order to evaluate the exploration ability of FSS, FSS-SAR and PSO. The first one, the L-Shape set of functions, can be defined as:

$$y = 10 \left\lceil \frac{max(|\mathbf{x}_i|)}{l} \right\rceil, \tag{15}$$

where $max(|\mathbf{x}_i|)$ is the maximum absolute value among the coordinates of position vector $\mathbf{x}_i$ and l is the width of the interval separating two consecutive plateaus given by $l =$

$\frac{x_{upperbound}}{d}$. d is the number of plateaus and $x_{upperbound}$ is the maximum value possible for every dimension.

These functions are composed by a number d of square steps centered in the origin. Figure 1 shows the surface defined in $[0;100]^2$ for L-Shape with 8 disks. The second

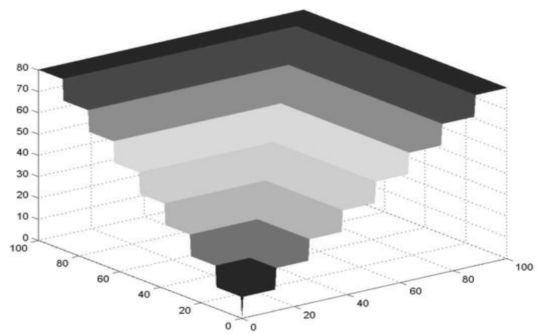

Figure 1. L-Shape function with $d = 8$

set of functions defined was C-Shape which can be defined as:

$$y = 10 \left\lceil \frac{radius(\mathbf{x}_i)}{l} \right\rceil, \tag{16}$$

where $radius(\mathbf{x}_i)$ is the euclidean distance between position vector $\mathbf{x}_i$ and the origin of the search space. l is the width of the interval separating two consecutive plateaus, given by $l = \frac{x_{upperbound}}{d}$. d is the number of plateaus and $x_{upperbound}$ is the maximum value possible for every dimension. C-Shape is composed by a number d of circular steps centered in the origin. Figure 2 shows the surface defined in $[0;100]^2$ for C-Shape with 8 disks.

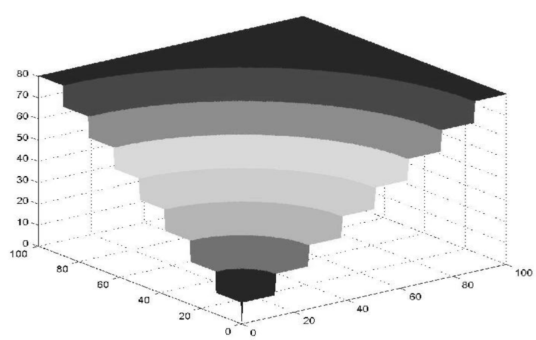

Figure 2. C-Shape function with $d = 8$

5. Tests and Results

In order to evaluate the performance of the three algorithms in the multi-plateaus search spaces, tests were performed in $[-100;100]^5$. For these tests, the initialization of all particles/fishes was made randomly but always in the last plateau. For SAR versions, α parameter was set to decay according to $\alpha = 0.8e^{-0.007*currentIteration}$.

The PSO version chosen for comparison is defined as in the work of Clerc and Kennedy [15]:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \chi[\mathbf{v}_i + c_1 r_1(\mathbf{pb}_i - \mathbf{x}_i(t)) + c_2 r_2(\mathbf{gb}_i - \mathbf{x}_i(t))], \tag{17}$$

where $\chi = \frac{2}{|2-(c_1+c_2)-\sqrt{(c_1+c_2)(c_1+c_2-4)}|}$ is known as constriction factor. For this version of PSO c_1 and c_2 values must be chosen satisfying $c_1 + c_2 \geq 4$ [15]. r_1 and r_2 are uniformly distributed random numbers in the interval $[0;1]$.

The parameters chosen for each algorithm are shown in table 1 based on the values utilized in the works of Bastos-Filho et al. [8] and Lima-Neto and Lacerda [12]. FSS Vanilla and original wFSS are referred to as FSS-V and wFSS.

TABLE 1. PARAMETERS

	W_{scale}	$Step_{ind}$	$Step_{vol}$
FSS-V	10^4	10	0.1
FSS-SAR	10^4	10	0.1
wFSS	500	10	0.1
wFSS-SAR	500	10	0.1
PSO	Version defined in [15]	$c_1 = c_2 = 2.05$	

Figures 3, 4, 5 and 6 represent the convergence curves for L-Shape and C-Shape functions with $d = 4$ and $d = 8$.

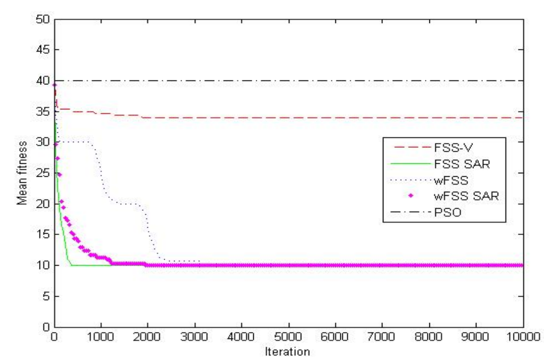

Figure 3. Convergence curves for L-Shape function with $d = 4$

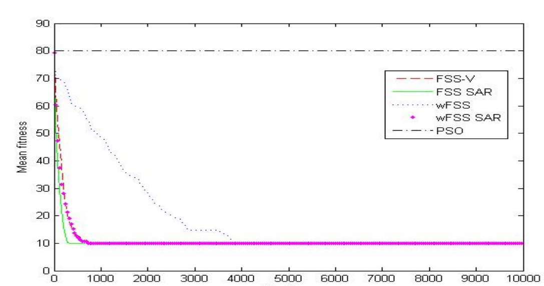

Figure 4. Convergence curves for L-Shape function with $d = 8$

From figure 3, it can be seen that PSO and FSS-V were trapped and could not achieve optimum values of fitness when solving L-Shape-4. SAR versions and wFSS were able to converge to optimum fitness values, however wFSS showed slow convergence. In the case of L-Shape-8 shown in figure 4, FSS-V was able to converge due to the reduction of plateaus width. This search space feature also had effect on the convergence of FSS-SAR, wFSS and wFSS-SAR. They all presented faster convergence than when solving L-Shape-4. PSO was not able to converge to optimum solutions in L-Shape-8.

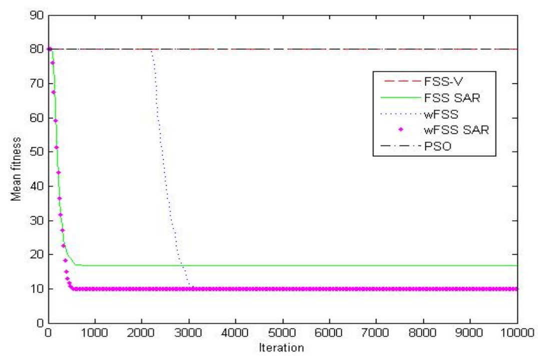

Figure 5. Convergence curves for C-Shape function with $d = 4$

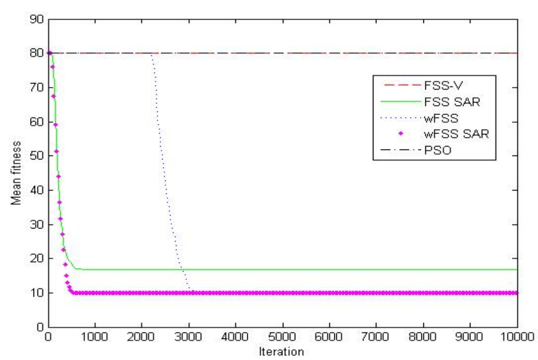

Figure 6. Convergence curves for C-Shape function with $d = 8$

For C-Shape functions, the convergence curves for C-Shape-4 in figure 5 show fast fitness improvement for both FSS-SAR and wFSS-SAR. However, FSS-SAR had fishes trapped in local optima is some of the test runs. Once more FSS-V and PSO could not improve in any test run and wFSS was able to optimally converge but in a slow manner

C-Shape-8 convergence analysis presented in figure 6 shows a similar behavior when compared to figure 5. PSO and FSS-V did not present fitness improvement, wFSS has shown slow convergence and FSS-SAR and wFSS-SAR were to find the global optima. However, FSS-SAR did not achieve the optimum values in all test runs as wFSS-SAR did.

TABLE 2. RESULTS FOR L-SHAPE AND C-SHAPE FOR $d = 4$ AND $d = 8$

	Algorithm	**Mean**	**Std Dev**	**Max**	**Min**
	FSS-V	40	0	40	40
	FSS-SAR	26	12.9943	40	10
C-Shape-4	wFSS	10	0	10	10
	wFSS-SAR	10	0	10	10
	PSO	40	0	40	40
	FSS-V	34	7.7013	40	10
	FSS-SAR	10	0	10	10
L-Shape-4	wFSS	10	0	10	10
	wFSS-SAR	10	0	10	10
	PSO	40	0	40	40
	FSS-V	80	0	80	80
	FSS-SAR	16.6667	20.3983	80	10
C-Shape-8	wFSS	10	0	10	10
	wFSS-SAR	10	0	10	10
	PSO	80	0	80	80
	FSS-V	10	0	10	10
	FSS-SAR	10	0	10	10
L-Shape-8	wFSS	10	0	10	10
	wFSS-SAR	10	0	10	10
	PSO	80	0	80	80

From table 2 it is possible to notice that FSS-V was only able to achieve optimum fitness values in L-Shape-8 and in some runs of L-Shape-4. FSS-SAR was able to reach minimum fitness in test functions, but had the fishes trapped in some runs in both C-Shape-4 and C-Shape-8. wFSS reached minimum fitness in all the cases tested, however the convergence analysis show that its convergence has occurred in a slow manner. wFSS-SAR presented the best results within the test functions considered. This algorithm has achieved optimum fitness in all the test runs and the convergence figures show a fast fitness improvement process.

Furthermore, tests were performed in functions Sphere, Rastrigin, Rosenbrock and Sum of Different Powers in order to demonstrate the performance of Stagnation Avoidance Routine versions on these benchmarking problems. The idea was to check if modifications proposed in FSS Vanilla and wFSS diminished their performance in functions different of the multi-plateau ones. The search spaces chosen are shown in table 3 and the parameters used are those presented in table 4. $Step_{ind}$ and $Step_{vol}$ are defined as percentages of the search space width. Table 5 registers the results achieved in 30 runs for each combination of function/algorithm.

TABLE 3. SEARCH SPACES

Function	**Search Space**
Sphere	$[-100; 100]^{30}$
Diff. Powers	$[-100; 100]^{30}$
Rastrigin	$[-5.12; 5.12]^{30}$
Rosenbrock	$[-100; 100]^{30}$

TABLE 4. PARAMETERS SET FOR BENCHMARKING FUNCTIONS TESTS

	W_{scale}	$Step_{ind}$	$Step_{vol}$
FSS-V	10^5	1%	0.05%
FSS-SAR	10^5	1%	0.05%
wFSS	500	1%	0.05%
wFSS-SAR	500	1%	0.05%
PSO	Version defined in [15]	$c_1 = c_2 = 2.05$	

TABLE 5. TESTS IN OTHER OBJECTIVE FUNCTIONS

Algorithm	Function	Mean	Std Dev	Min	Max
FSS-V	Sphere	0.9378	1.8681	0.0047	8.6538
FSS-SAR	Sphere	0.0006	0.0006	0.0000	0.0028
wFSS	Sphere	0.0050	0.0037	0.0000	0.0112
wFSS-SAR	Sphere	0.0067	0.0031	0.0000	0.0120
FSS-V	Diff. Power	2.8042	4.6763	0.0047	19.6687
FSS-SAR	Diff. Power	0.0012	0.0009	0.0001	0.0030
wFSS	Diff. Power	0.0015	0.0026	0.0000	0.0073
wFSS-SAR	Diff. Power	0.0034	0.0030	0.0000	0.0081
FSS-V	Rastrigin	222.6397	53.8456	116.6709	310.1582
FSS-SAR	Rastrigin	109.5423	17.1978	75.6461	139.3514
wFSS	Rastrigin	300.3256	21.8623	245.0616	338.4539
wFSS-SAR	Rastrigin	292.8820	23.3048	244.4116	342.5037
FSS-V	Rosenbrock	1490.4000	2351.1000	29.0758	10291.0000
FSS-SAR	Rosenbrock	31.7654	21.9400	23.8386	146.4889
wFSS	Rosenbrock	28.7706	0.0742	28.4032	28.8553
wFSS-SAR	Rosenbrock	28.7263	0.1906	27.9412	28.8385

From results presented in table 5 one could note that the Stagnation Avoidance Routine introduced in FSS did not have a negative impact in its ability to converge in search spaces other than multi-plateau ones, proposed in this paper. Further, the performance of FSS algorithm in Rosenbrock function, a valley-containing problem, and in Rastrigin funtion which has a multi-modal solution feature, also improved with the modification. The same could be observed with the bowl-shaped objective functions Sphere and Sum of Different Powers. In these functions, the performance of FSS-SAR was superior when compared to FSS-V. SAR modification did not presented neither a positive nor a negative impact in the niching FSS performance within the tests performed in the problems different of multi-plateaus.

6. Conclusion

Some classes of optimization problems such as integer and combinatorial programming originate very smooth and plateau containing search spaces. Many approaches try to use continuous metaheuristic procedures in order to tackle the aforementioned problems. In that case, the exploratory components of the search procedure utilized should be able to play a relevant role during the search process. Taking this fact into account, this work provided: (i) a set of functions which could be used for evaluating new techniques; (ii) a variation of the Fish School Search algorithm improving its exploratory ability.

The first contribution in this paper is the definition of new multi-plateau sets of functions. L-Shape and C-Shape functions can be used in order to evaluate algorithms performance in search spaces with a finite number of different levels. This kind of search space is common in Combinatorial Optimization family of problems, so that the functions proposed could be used for evaluating algorithms performance before trying to solve this sort of problem.

Furthermore, two new versions of Fish School Search algorithm were proposed with the introduction of the Stagnation Avoidance Routine in the individual component of the movement operator, the one responsible to promote a local search.

This modification was intended to increase the exploratory ability of the algorithm in the beginning of the search process. FSS-SAR and wFSS-SAR showed to be effective once these versions were able to quickly have fishes scaping from the highest plateau of multi-plateau functions, permitting faster convergence. PSO algorithm was not able to improve within the tests performed in multi-plateau functions.

Tests were performed in standard benchmark functions in order to check whether the modifications proposed diminished the performance of FSS in search spaces different from the plateau containing ones. The results showed that there is no performance reduction with the exploratory behavior introduced. On the contrary, this modification showed to be effective also in some of these test cases.

For future works, variations of the multi-plateau set of functions should be investigated. A multi-modal feature could be introduced in order to better simulate Combinatorial Optimization search spaces. Regarding modifications proposed in this paper in FSS algorithm, studies should be performed in order to evaluate the performance influence of the decay mode of the α parameter in SAR versions.

Acknowledgments

The authors thank to CAPES (Coordination for the Improvement of Higher-Education Personnel), Brazil for partial financial support for this paper.

References

[1] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Computing Surveys (CSUR)*, vol. 35, no. 3, pp. 268–308, 2003.

[2] M. Gendreau and J.-Y. Potvin, "Metaheuristics in combinatorial optimization," *Annals of Operations Research*, vol. 140, no. 1, pp. 189–213, 2005.

[3] M. Bellmore and G. L. Nemhauser, "The traveling salesman problem: a survey," *Operations Research*, vol. 16, no. 3, pp. 538–558, 1968.

[4] D. F. Barbosa, C. N. S. Junior, and A. é. Y. Kashiwabara, "Applying a variation of the ant colony optimization algorithm to solve the multiple traveling salesmen problem to route the teams of the electric power distribution companies," in *Proceedings of the annual conference on Brazilian Symposium on Information Systems: Information Systems: A Computer Socio-Technical Perspective-Volume 1*. Brazilian Computer Society, 2015, p. 4.

[5] H. M. Salkin and C. A. De Kluyver, "The knapsack problem: a survey," *Naval Research Logistics Quarterly*, vol. 22, no. 1, pp. 127–144, 1975.

[6] J. Błażewicz, W. Domschke, and E. Pesch, "The job shop scheduling problem: Conventional and new solution techniques," *European journal of operational research*, vol. 93, no. 1, pp. 1–33, 1996.

[7] V. S. Jorapur, V. S. Puranik, A. S. Deshpande, and M. Sharma, "A promising initial population based genetic algorithm for job shop scheduling problem," *Journal of Software Engineering and Applications*, vol. 9, no. 05, p. 208, 2016.

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

[9] C. J. A. Bastos-Filho and A. C. S. Guimarães, "Multi-Objective Fish School Search," *International Journal of Swarm Intelligence Research*, vol. 6, no. 1, pp. 23–40, 2015. [Online]. Available: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/ijsir.2015010102

[10] S. S. Madeiro, F. B. De Lima-Neto, C. J. A. Bastos-Filho, and E. M. Do Nascimento Figueiredo, "Density as the segregation mechanism in fish school search for multimodal optimization problems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6729 LNCS, no. PART 2, pp. 563–572, 2011.

[11] J. A. G. Sargo, S. M. Vieira, J. M. C. Sousa, and C. J. A. B. Filho, "Binary Fish School Search applied to feature selection: Application to ICU readmissions," *IEEE International Conference on Fuzzy Systems*, pp. 1366–1373, 2014.

[12] F. Buarque De Lima Neto and M. Gomes Pereira de Lacerda, "Weight based fish school search," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 270–277.

[13] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1. New York, NY, 1995, pp. 39–43.

[14] N. Hamta, S. F. Ghomi, F. Jolai, and M. A. Shirazi, "A hybrid pso algorithm for a multi-objective assembly line balancing problem with flexible operation times, sequence-dependent setup times and learning effect," *International Journal of Production Economics*, vol. 141, no. 1, pp. 99–111, 2013.

[15] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58–73, 2002.