

# Improved Multi-Objective Binary Fish School for Feature Selection

Mariana Macedo, Carmelo Bastos-Filho, Ronaldo Menezes<sup>†</sup>

Computer Engineering Department, University of Pernambuco, Recife, Brazil

<sup>†</sup>BioComplex Laboratory, School of Computing, Florida Institute of Technology, Melbourne, USA  
mgmm@ecomp.poli.br, carmelofilho@ieee.org, rmenezes@cs.fit.edu

## Abstract

The Multi-Objective Binary Fish School Search (MOBFSS) algorithm was proposed to solve optimization problems with two or three conflicting objectives and operating on discrete binary variables. The original proposal revealed good accuracy but it also exhibited a high computational cost. Here, we present strategies to obtain an improved version of MOBFSS that reaches lower Pareto fronts for minimization problems at a better computational cost. We also deploy local search procedures as proposed in BMOPSO-CDRLS to find solutions closer to the optimal solution. The achieved results outperform the state-of-art algorithms BMOPSO-CDR and BMOPSO-CDRLS in feature selection problems for hypervolume optimization. Hence, this paper contributes to the literature in Swarm Intelligence by introducing several algorithms that can be applied to improve feature selection in the context of classification programs.

## Introduction

Binary problems have been overlooked by the Swarm Intelligence community; the majority of the techniques are designed for continuous problems. Even though binary optimization problems have the same general issues present in continuous and discrete optimization, there are also significant differences. For instance, changes in one feature in a binary decision variable may completely change the context, which means that the selection of an feature can strongly impact performance. Hence, an optimization algorithm such as Fish School Search (FSS) needs to carefully select features. Furthermore, convergence operators may lead to populations of similar individuals. Such situation may happen in a continuous approach but are not as prominent as in binary problems. Consequently, new multi-objective binary algorithms should consider the idiosyncrasies of binary problems to avoid premature stagnation.

This paper proposes variations of Multi-Objective Binary Fish School Search (MOBFSS) considering the limitations and goals on binary optimization problems which include the minimize the number of features used in classification; we work with three datasets from UCI Machine Learning Repository (Lichman 2013) and look at feature selection as a function of classification error. In a nutshell, we present

strategies to obtain low computational cost avoiding an increase in the error classification.

## Multi-Objective Binary Fish School

The Multi-Objective Binary Fish School (MOBFSS) was proposed by Macedo et al (Macedo et al. 2017) to optimize problems with both conflicting objectives and binary decision variables. It was inspired on the manipulation of binary variables in BFSS (Sargo 2013; Sargo et al. 2014), the flip of multiple dimensions in IBFSS (Carneiro and Bastos-Filho 2016), and the operators for multi-objective problems in MOFSS (Bastos-Filho and Guimaraes 2015). In MOBFSS, the search process is similar to that of FSS in which a fish school moves in a limited search space (i.e., an aquarium) aiming at finding better regions. Each fish is a simple agent represented by a position vector  $x_i(t)$  and a weight  $w_i(t)$  associated with its performance  $f(x_i(t))$  through the iterations  $t$ . The movement of the school is described by three operators: individual, collective instinctive, and collective volitive.

In the individual operator, each fish choses a random position  $n_i(t)$  by flipping at most  $S_{ind}(t)$  dimensions, each with probability  $Flip_{ind}$ , and updates its weight as  $w_i(t+1) = w_i(t) + \Delta w_i(t)D_i(t)$ , where  $\Delta w_i(t)$  and  $D_i(t)$  are determined by the criteria of dominance as follows (Bastos-Filho and Guimaraes 2015). If the new solution dominates the current solution or it has a higher Crowding Distance (CD) in the case they are indifferent,  $\Delta w_i(t)$  is set to  $a$  or  $b$ , respectively, and the fish moves to the new position. Conversely, if the new solution is dominated by the current solution or it has a lower Crowding Distance (CD) in the case they are indifferent,  $\Delta w_i(t)$  is set to  $-a$  or  $-b$ , respectively, and the fish remains in its current position. The factor  $D_i(t)$  quantifies the extent to which a solution  $i$  is dominated by other solutions  $j$  while accounting for the number  $S_j(t)$  of solutions dominated by  $j$ , and it is calculated as  $D_i(t) = 1 - \frac{R_i(t)}{\max[R_i(t)]}$ , where  $R_i(t) = \sum_{j \in N, j \prec i} S_j(t)$ . Next, in the collective instinctive operator, every fish that successfully found a better position choses  $S_{inst}$  of their dimensions to set their values according to the values chosen by the majority. For instance, a fish will deliberately change the value of a specific dimension from 0 to 1 if most fish that improved their fitness also changed the value of that specific

dimension from 0 to 1.

In the collective volitive movement, each fish randomly chooses a leader from the External Archive as the guide. The External Archive is a repository that stores the best non-dominated solutions obtained through the search process. If the leader has better weight than the fish, then the dimensions that are different between the fish and the leaders are flipped. If the opposite happens, the dimensions that have equal values should be flipped instead. The step volitive  $S_{vol}$  constrains the maximum number of dimensions that can be flipped for each fish. Therefore, the collective volitive movement provides the expansion or contraction of the swarm. This operator is essential to the balance of exploration and exploitation during the optimization process.

The last operator is the *turbulence*, and its goal is to prevent the External Archive from getting stuck to a local minimum. In the turbulence, we perform a random flip into a copy of a solution randomly chosen within the External Archive. The original solution is replaced by the flipped one only if there is an improvement in the quality of the solution. In this case, an improvement means that the flipped solution dominates the original solution of the External Archive. The operator *turb* restricts how many features can be flipped.

### Novel Versions for the MOBFSS

We have introduced several versions of MOBFSS (henceforth called M-1) as below. Note that the M-1 version includes all the operators except for local search (see Table 1).

- **MOBFSS-2 (M-2):** M-2 was created to minimize the number of evaluations per iteration. The differences are: there is no evaluation of an individual movement, individuals always move; in the instinctive motion, the successful fish are those who improved their solution in the previous iteration. The hypothesis is that these simplifications can be done without impacting accuracy.
- **MOBFSS-1-LS (M-1-LS) and MOBFSS-2-LS (M-2-LS):** These are the versions of M-1 and M-2, respectively; they are endowed with local searches instead of turbulence. Given that local search strongly impacted in the improvement of BMOPSO-CDR (de Souza, Prudêncio, and Barros 2014), the hypothesis here is that that MOBFSS will also benefit.
- **MOBFSS-1-WO-T (M-1-WO-T):** M-1-WO-T removes the turbulence from M-1. The hypothesis is that any type of perturbation is fundamental to achieve good results.
- **MOBFSS-3 (M-3):** Similar to M-2 but with the individual operators excluded. As the individual operators mainly aims at providing diversity, the hypothesis is that the volitive movement is sufficient to provide diversity in binary problems.
- **MOBFSS-3-LS (M-3-LS):** Based on M-3 but with local search being applied instead of the turbulence operator. The hypothesis is that this operator will optimize the results.

Table 1 shows the operators utilized in each variation of the MOBFSS introduced in this paper. These new versions

Table 1: Operators of MOBFSS and their adoption in each version.

Operator	M-1	M-1-LS	M-1-WO-T	M-2	M-2-LS	M-3	M-3-LS
Individual Mov.	✓	✓	✓	✓	✓		
Instinctive Mov.	✓	✓	✓	✓	✓	✓	✓
Volitive Mov.	✓	✓	✓	✓	✓	✓	✓
Individual Eval.	✓	✓	✓				
Volitive Eval.	✓	✓	✓	✓	✓	✓	✓
Turbulence	✓			✓		✓	
Local Search		✓			✓		✓

were created aiming at improving the performance of the original version without the need to apply several operators and hence minimizing the complexity of the algorithm. Note that when an operator is removed, the requirements and initializations of some parameters for the algorithm become unnecessary.

### Experiments & Results

We performed the experiments using 2 computers: a MacBook Pro with a 3 GHz Intel Core i7 and 16 GB 1600 MHz DDR3, running macOS Sierra version 10.12.6; and a PC Intel Xeon 3.1 GHz, 16 GB RAM, arunning Ubuntu 15.10 64 bit. The programming language was Java but with the inclusion of the Jmetal library (Durillo and Nebro 2011); this library provides several techniques and metrics well established in the scientific community.

We chose the Support Vector Machine (SVM) as the classifier in our experiments because of its efficiency and robustness to several datasets and fields, and its results are good enough especially when combined with other preprocessing approaches (Hearst et al. 1998). Recall that we are comparing the MOBFSS versions according to their ability for doing feature selection, meaning that the MOBFSS versions select the features and use the SVM as a classifier using the features (i.e. the SVM is the *de facto* fitness function in our approach). We selected 3 datasets from UCI repository (Lichman 2013). They are Wine, Ionosphere, and Sonar with 13, 34, and 60 features, respectively. The number of features makes them more or less complex; the higher the number of features the more complex is the problem. Each version and dataset requires a different feature configuration. However, as those techniques aim to solve a large number of problems, we performed a parametric analysis for each one of the three datasets, and we chose the best solution overall.

We executed 30 simulations for each experiment in each dataset. The stop criterion and the initialized values were adopted as proposed by Souza et al. (de Souza et al. 2011), where the maximum number of evaluations of the fitness function is 200,000. The variations of the MOBFSS we propose here are compared to the BMOPSO-CDR and BMOPSO-CDRLS.

Tables 2 and 3 presents the performance of each variation as well as the two aforementioned PSO versions. The tables include the mean execution time plus the standard deviation, maximum, and minimum values for three datasets used.

The Wine dataset is the easiest experiment in this paper. Because of its simplicity, the utilized configuration rarely

Table 2: Execution time (in seconds) for 30 trials of each dataset.

Dataset	Algorithm	Mean	Standard Deviation	Max	Min
Wine	<i>BMOPSO – CDR</i>	2899.57	149.25	3428.57	2562.58
	<i>BMOPSO – CDRLS</i>	2621.05	148.98	3020.96	2246.30
	<i>MOBFSS – 1 – WO – T</i>	2057.43	362.89	2727.51	1274.05
	<i>MOBFSS – 1 – LS</i>	2052.50	311.39	2545.47	1279.34
	<i>MOBFSS – 1</i>	2502.81	1010.05	7004.17	1286.85
	<i>MOBFSS – 2</i>	1739.95	646.60	4122.74	667.47
	<i>MOBFSS – 2 – LS</i>	1597.49	619.32	3885.47	826.13
	<i>MOBFSS – 3</i>	860.17	186.76	1306.56	567.42
	<i>MOBFSS – 3 – LS</i>	<b>791.13</b>	171.04	1296.22	563.56
Ionosphere	<i>BMOPSO – CDR</i>	2226.35	188.65	2857.75	1978.61
	<i>BMOPSO – CDRLS</i>	2059.85	160.30	2543.17	1847.74
	<i>MOBFSS – 1 – WO – T</i>	2890.87	794.07	5997.38	2251.15
	<i>MOBFSS – 1 – LS</i>	2553.60	388.01	3819.58	2014.57
	<i>MOBFSS – 1</i>	2689.21	442.26	4171.33	2283.17
	<i>MOBFSS – 2</i>	2580.94	479.56	3923.67	1970.44
	<i>MOBFSS – 2 – LS</i>	2191.95	546.98	4566.73	1907.93
	<i>MOBFSS – 3</i>	<b>1865.23</b>	308.49	3349.52	1641.54
	<i>MOBFSS – 3 – LS</i>	1946.28	506.94	3716.42	1607.20
Sonar	<i>BMOPSO – CDR</i>	2265.14	41.18	2354.69	2180.68
	<i>BMOPSO – CDRLS</i>	2104.58	169.48	2699.18	1944.72
	<i>MOBFSS – 1 – WO – T</i>	2090.05	198.41	2543.17	1847.74
	<i>MOBFSS – 1 – LS</i>	2528.43	169.47	2887.85	2150.29
	<i>MOBFSS – 1</i>	2821.31	194.26	3245.92	2449.60
	<i>MOBFSS – 2</i>	2575.69	53.04	2700.62	2453.51
	<i>MOBFSS – 2 – LS</i>	1586.84	127.07	1915.93	1496.09
	<i>MOBFSS – 3</i>	1393.04	124.82	1630.84	1191.05
	<i>MOBFSS – 3 – LS</i>	<b>1337.24</b>	114.36	1541.09	1195.842

impacts the results. Several settings and algorithms achieve the same best solutions, and 100 iterations are enough to reach them. In Table 2, M-3-LS shows the shortest execution time in seconds. Given the variations reaches similar solutions (see Figure 1), it is recommended to utilize the fastest version, M-3-LS. Using less than six features, the error classification for Wine remains less than 0.011%, while one of the features is of particular importance for providing high accuracy.

The Ionosphere classification is a harder problem than the classification in Wine, and each algorithm presented different results. The M-3 and M-3-LS variations had the fastest execution time as displayed in Table 2. In Table 3 we observe that the maximum hypervolume (HV) was obtained by M-3-LS which is also the second fastest algorithm. In addition, BMOPSO-CDRLS, M-1-LS, M-1 and M-3-LS shows similar mean values which indicates that the usage of local search is positive, but M-1 was not strongly impacted. Although the turbulence and local search were not determinant in this dataset, their usage seems positive for the algorithms. Spacing (S) can be evaluated to identify the diversity of the Pareto Fronts. In Table 3, S is optimized by M-3, but its performance is not one of the best. Looking at the HV and S, it is possible to observe that the M-3-LS is the best option because the diversity is easier in suboptimal locations. Table 3 shows that the maximization of the Maximum Spread (MS) is accomplished by M-3-LS revealing a considerable extension of the Pareto front, and longer Pareto fronts are easier to achieve in suboptimal locations.

The statistical test for Ionosphere dataset is exhibited in Table 4. In this table,  $\blacktriangle$  means that the row is better than the column,  $\nabla$  represents that the row is worse than the column, and  $-$  means that the results are similar. The statistical tests were done using the Wilcoxon test with a 95% of the confidence interval.

M-3-LS statistically beats the other algorithms, and it

Table 3: Mean, Standard Deviation, Maximum and Minimum of the 30 simulations of each algorithm calculated by the three metrics of Ionosphere and Sonar datasets.

Dataset	Algorithm	Mean	Standard Deviation	Max	Min
Ionosphere (Hypervolume)	<i>BMOPSO – CDR</i>	0.8066	0.0182	0.8390	0.7639
	<i>BMOPSO – CDRLS</i>	0.8805	0.0181	0.8918	0.8126
	<i>MOBFSS – 1 – WO – T</i>	0.8592	0.0222	0.8931	0.7925
	<i>MOBFSS – 1 – LS</i>	0.8784	0.0226	<b>0.9172</b>	0.8453
	<i>MOBFSS – 1</i>	0.8759	0.0127	0.8960	0.8468
	<i>MOBFSS – 2</i>	0.8074	0.0272	0.8675	0.7632
	<i>MOBFSS – 2 – LS</i>	0.8084	0.0645	0.8793	0.7325
	<i>MOBFSS – 3</i>	0.8368	0.0140	0.8700	0.8160
	<i>MOBFSS – 3 – LS</i>	<b>0.8895</b>	0.0197	0.9066	0.8435
Sonar (Hypervolume)	<i>BMOPSO – CDR</i>	0.6926	0.0105	0.7151	0.6687
	<i>BMOPSO – CDRLS</i>	0.8019	0.0045	0.8115	0.7941
	<i>MOBFSS – 1 – WO – T</i>	0.8306	0.0152	0.8678	0.8086
	<i>MOBFSS – 1 – LS</i>	<b>0.8608</b>	0.0031	0.8676	0.8558
	<i>MOBFSS – 1</i>	0.8348	0.0137	0.8639	0.8126
	<i>MOBFSS – 2</i>	0.7487	0.0204	0.7887	0.7145
	<i>MOBFSS – 2 – LS</i>	0.7974	0.0066	0.8160	0.7833
	<i>MOBFSS – 3</i>	0.7844	0.0151	0.8243	0.7605
	<i>MOBFSS – 3 – LS</i>	0.8477	0.0045	0.8557	0.8367
Ionosphere (Spacing)	<i>BMOPSO – CDR</i>	0.0327	0.0130	0.0692	0.0095
	<i>BMOPSO – CDRLS</i>	0.0731	0.0221	0.1095	0.0128
	<i>MOBFSS – 1 – WO – T</i>	0.0278	0.0349	0.1396	0.0014
	<i>MOBFSS – 1 – LS</i>	0.0344	0.0232	0.0880	0.0014
	<i>MOBFSS – 1</i>	0.0329	0.0180	0.0624	0.0023
	<i>MOBFSS – 2</i>	0.0351	0.0183	0.1042	0.0043
	<i>MOBFSS – 2 – LS</i>	0.0862	0.0596	0.1676	0.0118
	<i>MOBFSS – 3</i>	<b>0.0140</b>	0.0064	0.0314	0.0022
	<i>MOBFSS – 3 – LS</i>	0.0877	0.0392	0.1414	0.0102
Sonar (Spacing)	<i>BMOPSO – CDR</i>	0.0247	0.01131	0.0561	0.0090
	<i>BMOPSO – CDRLS</i>	0.1136	0.0230	0.1848	0.0760
	<i>MOBFSS – 1 – WO – T</i>	0.0189	0.0163	0.0574	0.0036
	<i>MOBFSS – 1 – LS</i>	0.0581	0.0073	0.0874	0.0508
	<i>MOBFSS – 1</i>	0.0210	0.0144	0.0606	0.0052
	<i>MOBFSS – 2</i>	<b>0.0140</b>	0.0169	0.0728	0.0000
	<i>MOBFSS – 2 – LS</i>	0.1832	0.0330	0.2392	0.1184
	<i>MOBFSS – 3</i>	0.0177	0.016	0.0692	0.0013
	<i>MOBFSS – 3 – LS</i>	0.1367	0.022	0.1904	0.0965
Ionosphere (Maximum Spread)	<i>BMOPSO – CDR</i>	0.8654	0.0197	0.8977	0.8274
	<i>BMOPSO – CDRLS</i>	0.8785	0.0220	0.9258	0.8414
	<i>MOBFSS – 1 – WO – T</i>	0.8990	0.0344	1.0093	0.8418
	<i>MOBFSS – 1 – LS</i>	0.8926	0.0232	0.9443	0.8380
	<i>MOBFSS – 1</i>	0.8874	0.0217	0.9372	0.8475
	<i>MOBFSS – 2</i>	0.9086	0.0240	0.9618	0.8747
	<i>MOBFSS – 2 – LS</i>	0.9185	0.0217	0.9801	0.8814
	<i>MOBFSS – 3</i>	0.9035	0.0130	0.9272	0.8720
	<i>MOBFSS – 3 – LS</i>	<b>0.9314</b>	0.0248	0.9843	0.8889
Sonar (Maximum Spread)	<i>BMOPSO – CDR</i>	0.8975	0.0208	0.9567	0.8601
	<i>BMOPSO – CDRLS</i>	0.9373	0.0272	0.9942	0.8802
	<i>MOBFSS – 1 – WO – T</i>	0.9208	0.0100	0.9418	0.9043
	<i>MOBFSS – 1 – LS</i>	0.9338	0.0173	0.9869	0.9023
	<i>MOBFSS – 1</i>	0.9174	0.0134	0.9419	0.8858
	<i>MOBFSS – 2</i>	0.9641	0.0159	0.9880	0.9294
	<i>MOBFSS – 2 – LS</i>	0.9647	0.0514	1.0372	0.7885
	<i>MOBFSS – 3</i>	0.9468	0.0147	1.0014	0.9239
	<i>MOBFSS – 3 – LS</i>	<b>0.9985</b>	0.0211	1.0322	0.9518

is one of the fastest algorithms. However, M-1-LS could reach the best value of HV (the best solution for the problem). The algorithms displayed in Table 4 are better than the algorithms BMOPSO-CDR, M-2, M-2-LS and M-3. Figure 1 presents the best solution found by the 30 trials. M-1-LS displayed a bigger and better Pareto front than M-3-LS; these versions also show a better Pareto front than BMOPSO-CDRLS.

Last, the Sonar dataset is the most complex experiment performed in this paper because the number of features is larger highlighting the efficiency of the MOBFSS versions. The execution time of all the algorithms displayed in Table 2 is minimum in M-3-LS. However, the best algorithm regarding performance is M-1-LS (as shown in Table 3). Between those 2 versions, we recommended the use of the version which prioritizes the main goal of a problem. Some problems will require higher accuracy, and others are better to achieve fast answers. For this dataset, the usage of the local search was positive in all the cases. Moreover, when comparing the MOBFSS versions without local search to the BMOPSO-CDRLS, the MOBFSS versions were better. The

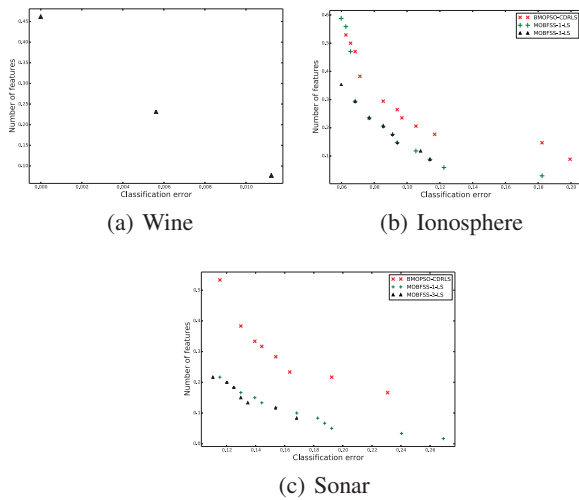


Figure 1: Pareto front for the best results in the External Archive (EA).

diversity was minimized by M-2, but, as mention before, it is easier to achieve better diversity with suboptimal solutions. Thus, the algorithms which achieve better solutions provide worse results for S. M-3-LS presented the best values for the extension found on Pareto fronts (MS).

Table 4: Comparison of the best algorithms for the performance of Ionosphere dataset.

	BP-CDRLS	MF-1	MF-1-WO-T	MF-1-LS	MF-3-LS
BP-CDRLS		—	—	—	▽
MF-1	—		▲	—	▽
MF-1-WO-T	▽	▽		▽	▽
MF-1-LS	—	—	▲		—
MF-3-LS	▲	▲	▲	—	

Table 5: Comparison of the best algorithms for the performance of Sonar dataset.

	BP-CDRLS	MF-1	MF-1-LS	MF-3-LS
BP-CDRLS		▽	▽	▽
MF-1	▲		▽	▽
MF-1-LS	▲	▲		▲
MF-3-LS	▲	▲	▽	

Table 5 shows the pairwise comparison of the versions using the same Wilcoxon test used before. M-1-LS is statistically better than the others, and M-3-LS also beats the others except for M-1-LS. In addition, BMOPSO-CDRLS is better than BMOPSO-CDR, M-2, M-2-LS and M-3. Figure 1 depicts the best solution for the 30 trials where clearly the best solution was achieved by M-3-LS. It is also important to note that the usage of 10-20% of the dimensions yields a 10-17% in error classification.

## Conclusion

We proposed new versions for MOBFSS aiming to minimize the computational cost while maintaining performance. A new lighter version called MOBFSS-3-LS presents a good accuracy with reduced computational cost, but it did not beat MOBFSS-1-LS regarding hypervolume in all databases. The algorithms also aimed to avoid local minima, and the local search showed significant efficiency in that regard. We contributed to Swarm Intelligence by suggesting new binary multi-objective algorithm variations in particular to the problem of feature selection. We suggest that our variations need to be analyzed considering other configurations, and tested in real-world problems.

## References

- Bastos-Filho, C. J., and Guimaraes, A. C. 2015. Multi-objective fish school search. *International Journal of Swarm Intelligence Research (IJSIR)* 6(1):23–40.
- Carneiro, R. F., and Bastos-Filho, C. J. 2016. Improving the binary fish school search algorithm for feature selection. In *Computational Intelligence (LA-CCI), 2016 IEEE Latin American Conference on*, 1–6. IEEE.
- de Souza, L. S.; de Miranda, P. B.; Prudencio, R. B.; and Barros, F. d. A. 2011. A multi-objective particle swarm optimization for test case selection based on functional requirements coverage and execution effort. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, 245–252. IEEE.
- de Souza, L. S.; Prudêncio, R. B.; and Barros, F. d. A. 2014. A hybrid binary multi-objective particle swarm optimization with local search for test case selection. In *Intelligent Systems (BRACIS), 2014 Brazilian Conference on*, 414–419.
- Durillo, J. J., and Nebro, A. J. 2011. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software* 42(10):760–771.
- Hearst, M. A.; Dumais, S. T.; Osuna, E.; Platt, J.; and Scholkopf, B. 1998. Support vector machines. *IEEE Intelligent Systems and their applications* 13(4):18–28.
- Lichman, M. 2013. Uci machine learning repository.
- Macedo, M. G. d. M.; Bastos-Filho, C. J. A.; Vieira, S. M.; and Sousa, J. M. C. 2017. Multi Objective Binary Fish School Search. *Critical Developments and Applications of Swarm Intelligence*.
- Sargo, J. A.; Vieira, S. M.; Sousa, J. M.; and Bastos-Filho, C. J. 2014. Binary fish school search applied to feature selection: Application to icu readmissions. In *FUZZ-IEEE*.
- Sargo, J. A. G. 2013. Binary fish school search applied to feature selection.

## Acknowledgments

Mariana Macedo thanks to University of Pernambuco (PFA-UPE) and CAPES (Coordination for the Improvement of Higher-Education Personnel) for the partial financial support provided. Ronaldo Menezes was partially supported by the US Army Research Office under Agreement Number W911NF-17-1-0127.