BENCHMARKING BIO-INSPIRED COMPUTATION ALGORITHMS AS WRAPPERS FOR FEATURE SELECTION

Dražen BAJER, Bruno ZORIĆ, Mario DUDJAK, Goran MARTINOVIĆ
Department of Software Engineering, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Josip Juraj Strossmayer University of Osijek, kneza Trpimira 2b, 31000 Osijek, tel. +38531495415, E-mail: mario.dudjak@ferit.hr

ABSTRACT

Reducing the number of features when applying machine learning algorithms may be beneficial not only from the standpoint of computational cost but also of overall quality. Wrapper-based procedures are widely utilised to achieve this. The choice of the wrapper is of utmost importance. Bio-inspired computation algorithms represent a viable choice and are widely adopted. Due to the sheer number of available algorithms, this choice could prove to be somewhat difficult, especially since not all are made equally. The aim of this paper is to explore several optimisers on diverse datasets representing classification problems in order to evaluate their performance and suitability for the task of feature selection.

Keywords: *bio-inspired computation, classification, dimensionality reduction, feature selection, wrapper model*

1. INTRODUCTION

The rising numbers in both feature and sample count evident in datasets becoming available today render the analysis of such data ever more challenging. The large number of features is arguably one of the more prominent factors associated with this difficulty since not all are relevant and useful. In many instances the dimensionality reduction that can be achieved by removing redundant and irrelevant features is essential not only for creating simpler models with less computational resources, but also for increasing model accuracy and interpretability. In that light, feature subset selection or simply feature selection (FS) has become an indispensable pre-processing step for many tasks in machine learning, such as classification, clustering and regression. It can be conducted in a variety of ways and numerous approaches have been proposed (see, e.g., [1, 2] for an overview). It certainly represents a valuable tool with applications across a wide range of problem domains where machine learning is employed for data analysis and as decision support. For example, a general overview of feature selection applications can be seen in [3], whilst an overview specifically related to the area of medicine can be seen in [4].

As stated earlier, the main driving force behind the FS task becoming more challenging is the growth in dataset feature count. Hence, a complete enumeration of all $2^m - 1$ subsets is infeasible even for a moderate feature count (m) [1, 3]. Therefore, a need for efficient search strategies is self-evident. Some review or survey studies (see, e.g., [1, 3, 5]) already found evolutionary and swarm intelligence algorithms (can be regarded as subsets of bio-inspired computation) to be promising approaches for obtaining feature subsets contributing to the overall performance. The principal reason is their ability to effectively explore large portions of the search space ($S = \{0,1\}^m$ for FS). No specific properties of that space assumed, either. A plethora of such algorithms have been proposed and many of them have been (successfully) applied to FS. Most of these algorithms are used to compose the wrapper model [5], which usually yields smaller and better subsets (in terms of model performance) than the filter model, but is typically computationally more expensive [2, 4].

The research on bio-inspired (or nature-inspired) computation is extremely active [6] and new algorithms are proposed on an almost daily basis. Many claim or promise superior performance and ease of use (few user-defined parameters, simple algorithm structure, etc.). Whether some approaches should be preferred when solving a given problem is a question that naturally comes to mind. In an attempt to answer it, in this paper the performance of several such algorithms is investigated on the problem of feature selection. More precisely, wrapper-based FS for classification needs was tackled with some well-established algorithms and other more recent ones. The experimental comparison was conducted on diverse datasets in order to offer an insight into the above-mentioned. Several indicators were recorded and the obtained results were statistically analysed to this end. The aim being to gain a glimpse into their differences but also to highlight some of their advantages and shortcomings, at least with respect to their suitability for the task of FS.

The rest of the paper is organised as follows. A concise overview of feature selection and bio-inspired computation is given in Sect. 2. In Sect. 3 two important elements are considered that need particular attention when bio-inspired optimisers are applied to feature selection. The experimental setup and the obtained results are reported and discussed in Sect. 4. Finally, the drawn conclusions and some possible directions for future work are stated in Sect. 5.

2. PRINCIPAL TERMS AND IDEAS

Feature selection as a significant tool in machine learning is traditionally approached in a couple of distinct ways – filter and wrapper methods. Bio-inspired computation algorithms are frequently employed as wrappers but their application warrants consideration of two elements – solution representation and evaluation. The principal terms and ideas regarding both of the aforementioned are established next.

2.1. Feature selection

Working with data described by a large number of features is challenging from various perspectives. Unfortunately, a large number of features does not imply a good description of the considered phenomenon since unnecessary (irrelevant and redundant) features are commonly present [7]. Basically, the FS task is to remove such features. Usually, however, the features cannot be examined independently due to their interactions. These can vary in complexity, meaning that a useless single feature in combination with one or more others could become significant for the target concept [8].

Based on the evaluation procedure, various methods for approaching the FS task are available. These are traditionally categorised as wrappers and filters. Filters rely solely on intrinsic data properties, utilising for example ideas from information theory (e.g. information gain), and require no knowledge regarding the classifier which makes them comparatively fast [9]. They rank features according to these properties, where a predetermined number of them is then considered for modelling the problem. Hence, filters are often applied as a preprocessing step. Wrappers on the other hand are guided by model performance. They are essentially search algorithms with an incorporated classifier (hence the name, "wrapping" around a classifier). Due to their reliance on the output of a certain model, they are computationally more expensive and occasionally prone to overfitting. However, they are generally able to discover more complex relationships between features, and often yield better (higher model performance) and smaller subsets (less selected features). Both of the aforementioned can be combined resulting in hybrid approaches aimed at alleviating the drawbacks of either singular approach type.

2.2. Bio-inspired computation

Roughly speaking, bio-inspired algorithms represent population-based global optimisation methods. Generally, they incorporate some form of variation operators on population members (candidate solutions) to create new solutions i.e. to sample the search space and, typically, selection to drive the search towards promising regions of that space. They can be considered stochastic, derivative-free methods and as such they rely solely on the objective function values associated with points of the search space (black-box optimisation). Some are better at performing a coarse-grained search (exploration) than a fine-grained search around promising points of the search space (exploitation) and vice-versa. It is, however, well-known that achieving a balance between those two search aspects is a key ingredient for achieving high and consistent performance.

Although there are some well-established and particularly popular optimisers, like genetic algorithms (GAs) [10], particle swarm optimisation (PSO) [11] and differential evolution (DE) [12], the list of bio-inspired algorithms available in the literature is immense (see, e.g., [13] for a comprehensive, albeit incomplete list) and steadily growing. The trend of proposing new optimisers is clearly apparent in the literature. Peculiar metaphors are often introduced to describe the search mechanism(s). Such an approach to algorithm design and development (referred to as metaphor-centric in [14]) gained some criticism (see, e.g., [6, 14, 15, 16]) since the majority claims to be "novel" and "superior" in many aspects to previous ones, without offering proper supporting evidence. Nevertheless, some grew popular and have found diverse applications. Certainly, there are distinctions amongst many of these algorithms, which are mostly reflected in the operators for creating new solutions i.e. for sampling the search space. Interestingly, the vast majority of new proposals have operators defined in the continuous domain ($\mathbb{R}^m$). These operators typically do not use dedicated probability distributions for generating perturbations but use scaled differences between available candidate solutions kept as population members or as separate entities (popularised mainly by PSO and DE). Further, from the viewpoint of a practitioner, the number of user-defined algorithm parameters is important since tuning is typically necessary for attaining best performance on the problem at hand. In that regard, fewer parameters might be considered a better option. There are a number of algorithms (for the most part more recent ones) that have a single user-defined parameter. As a rule, this parameter represents the population size. However, this lack of more extrinsic parameters is typically achieved by incorporating one or more intrinsic/internal parameters that are either fixed, randomly generated or dynamically adjusted (see, e.g., [17, 18, 19]). In other words, this can be considered a deliberate attempt to hide algorithm control parameters that might influence their behaviour. Regardless, this might prove to be a limiting factor in terms of flexibility and versatility compared to algorithms with multiple tunable parameters and should be kept in mind. With all that being said, it not difficult to imagine the trouble involved in making an appropriate choice of optimiser for a practitioner without expertise in the field of bio-inspired computation.

3. BIO-INSPIRED COMPUTATION FOR WRAPPER-BASED FEATURE SELECTION

Tackling any problem with bio-inspired optimisers requires that an appropriate solution representation and evaluation criterion is chosen. The task of FS is no different in that regard. Various options are available for these two requirements.

A straightforward and intuitive solution representation is a binary vector $\mathbf{b} = (b_1, \ldots, b_m) \in \{0,1\}^m$, where $b_i = 1$ or 0 for $i = 1, \ldots, m$ indicates that the i-th feature is or is not selected, respectively. Although this representation fits GAs, it does not fit the majority of other algorithms that have operators defined in the continuous domain. Hence, a transformation to obtain binary vectors is necessary. A simple and widely used approach is the application of a sigmoid function (e.g. the logistic function) followed by thresholding. The purpose of the sigmoid function is to map the solution components to the same interval (typically $[0,1]$ or $[-1,1]$). This may also be achieved by introducing a predefined boundary on the search space (e.g. $[0,1]^m$) along with bound constraint handling, which is an additional design variable that can influence algorithm be-

haviour. It must be noted that these binary vectors are only generated prior to evaluation and do not replace the original, real-valued solution vectors. The drawback to such a transformation is the many-to-one mapping. Consequently, real-valued vectors in the population that are in close proximity are likely to result in identical binary vectors, which is also likely to happen with solutions created by combining them. A few other representations can be found in the literature (see, e.g., [5]). Some of these require that the number of selected features is fixed, which implies a priori knowledge or decision making is necessary.

As mentioned earlier, in the wrapper model, the classifier performance is used as the quality metric for found feature subsets. The classifier is used as a black-box and any classifier of choice may be employed. Often a simple and fast classifier is chosen as to keep the computational overhead as small as possible. A number of measures for quantifying the classifier performance are available and making the proper choice is important and requires care. Commonly used is the classification accuracy (CAC) or conversely, the misclassification-rate (MCR). This is not surprising since both are intuitive and easy to calculate. Moreover, often a linear combination of this measure and a penalty term is employed for the evaluation of subsets (see, e.g., [20, 21, 22]). Typically, the penalty term is represented by the normalised subset size. This represents a simple approach to the treatment of FS as a bi-objective problem (these two objectives are not always conflicting [5]), where the contribution of the model performance and subset size is determined by the pre-set weight. Determining this weight is a major issue of that approach to multi-objective optimisation and is subjective (although as a rule, emphasis is put on model performance). It is important to remark that the CAC or MCR as a measure of classification performance can be misleading in the case of class imbalance [23], which could subsequently lead to the loss of features relevant to the minority class(es). Hence, a more sensible measure, like the F1-score or the geometric mean of trues, might be better suited [24] and should arguably be preferred.

4. EXPERIMENTS AND RESULTS

In order to assess the performance and suitability of some bio-inspired computation algorithms for the wrapper-based FS task, an experimental analysis was conducted on a test bed comprised of diverse datasets. The employed datasets are concisely given in Table 1. All datasets have been taken from the UCI repository [25], except A_5 which has been taken from the KEEL repository [26]. The algorithms selected for the purpose of the above-mentioned along with the utilised parameter configurations are as follows:

- Artificial bee colony algorithm (ABC) [27, 28] – population size $SN = 30$ and $limit = 250$.
- Differential evolution (DE) [12] – population size $NP = 50$, scale factor $F = 0.5$ and crossover-rate $CR = 0.9$
- Simple genetic algorithm (SGA) [10] – population size $N = 50$, crossover probability $p_c = 0.9$ and mutation probability $p_m = 0.1$
- Global best particle swarm optimisation (PSO) [11] – population size $NS = 30$, acceleration coefficients $c_1 = c_2 = 1.496$ (for the cognitive and social component, respectively) and inertia weight $\omega = 0.7298$
- Sine cosine algorithm (SCA) [17] – population size $n = 30$
- Whale optimisation algorithm (WOA) [18] – population size $n = 30$
- Jaya algorithm [19] – population size $n = 20$

Amongst the selected algorithms, the SGA and DE can be categorised as evolutionary algorithms, whereas the others as swarm intelligence algorithms. It should be noted that only standard algorithm variants have been considered since the goal was to offer an insight into the differences of their intrinsic elements. The algorithm parameters have been set by following recommendations found in the literature. As can be observed, the three most recent of the considered algorithms (SCA, WOA and Jaya) expose only a single user-defined parameter – the population size, whereas the others require that two or even more parameters are set.

Table 1 Characteristics of the datasets comprising the test bed

A	name	#features	#samples	#classes
1	QSAR biodegradation	41	1055	2
2	Connectionist Bench	60	208	2
3	Hill-Valley	100	1212	2
4	Ionosphere	34	351	2
5	Sonar	60	208	2
6	Dermatology	34	358	6
7	Image Segmentation	19	210	7
8	Libras Movement	90	360	15
9	Musk (Version 1)	166	476	2
10	Parkinsons	22	195	2
11	Statlog (Vehicle Silhouettes)	18	846	4
12	LSVT Voice Rehabilitation	310	126	2
13	Urban Land Cover	147	675	9
14	Wine	13	178	3

4.1. Methodology and setup

The only of the considered algorithms that operates in the binary search space is the SGA. For the others, the solutions were constrained into $[0,1]^m \subset \mathbb{R}^m$ and binary vectors were created prior to evaluation via threshholding i.e. for each real-valued vector $\mathbf{v} = (v_1, \ldots, v_m) \in [0,1]^m$ a separate binary vector $\mathbf{b}$ was created as

$$b_i = \begin{cases} 1 & \text{if } v_i < \theta, \\ 0 & \text{otherwise} \end{cases} \quad i = 1, \ldots, m, \tag{1}$$

where the threshold value was set to $\theta = 0.5$. Bound constraint handling was performed by resetting solution components outside the search space to the nearest boundary value (this only applies to the algorithms operating in the real domain). Feature subsets were evaluated by the F1-score, attained after applying the nearest neighbour (1-NN) classifier (the same approach as in [29]).

For each algorithm and dataset combination 25 independent runs were performed. The termination criterion was

the execution of the pre-set maximal number of function evaluations, $\text{NFEs}_{\max} = 10000$, in order to enable a fair comparison. Population initialisation was conducted uniformly at random inside the whole search space ($\{0,1\}^m$ for the SGA and $[0,1]^m$ for the other algorithms). Further, stratified holdout evaluation was employed. Accordingly, the standard split ratio of $0.5:0.25:0.25$ was used for training, validating and testing, respectively. A single split was generated and used in all algorithm runs since the primary goal was to evaluate the optimisers in terms of performance and stability. Feature scaling, via normalisation into the $[0,1]$ range, was performed as a pre-processing step on each dataset in order to mitigate the influence of varying value ranges.

4.2. Results and discussion

The obtained results in terms of classification performance on the test subsets are presented in Table 2. In order to ease the determination of the mutual order of the competing algorithms in terms of overall performance, several distinct measures derived from the average F1-scores across all datasets are provided (a similar approach as used in [30]). Denoted by FR are the average rankings obtained by applying the Friedman test for multiple comparisons. Denoted by d_2 are the Euclidean distances from a hypothetical perfect classifier (achieves an F1-score of 1 on all datasets). Moreover, the Chebyshev distance (d_∞) is provided for breaking ties. In all of the aforementioned cases, a lower value is better. Additionally, the number of times that an individual algorithm achieved the best and worst average F1-score on particular datasets are listed in the table. To facilitate readability, values representing the best are given in boldface.

Table 2 Overall results in terms of measures derived from the average F1-scores across all datasets

Alg.	FR	d_2	d_∞	#Best	#Worst
ABC	3.96	0.79	0.42	2	1
DE	**1.71**	**0.76**	**0.40**	**9**	**0**
SGA	4.61	0.80	0.42	1	1
PSO	5.14	0.81	0.42	0	4
SCA	4.71	0.83	0.41	2	7
WOA	5.07	0.82	0.41	0	1
Jaya	2.79	0.78	0.41	1	**0**

Table 3 Ranks obtained from the Wilcoxon test for multiple comparisons

	ABC	DE	SGA	PSO	SCA	WOA	Jaya
ABC	—	18.0$^\diamond$	60.0	71.0	75.0	68.0	22.5$^\diamond$
DE	87.0†	—	91.0†	105.0†	98.0†	104.0†	91.0†
SGA	31.0	14.0$^\diamond$	—	66.0	70.0	64.0	17.0$^\diamond$
PSO	34.0	0.0$^\diamond$	39.0	—	65.0	58.0	4.0$^\diamond$
SCA	30.0	7.0$^\diamond$	35.0	40.0	—	40.0	16.0$^\diamond$
WOA	37.0	1.0$^\diamond$	41.0	47.0	65.0	—	4.0$^\diamond$
Jaya	82.5	14.0$^\diamond$	88.0†	101.0†	89.0†	101.0†	—

Even a brief glance at the results reveals that DE stands out in terms of performance and that it is directly followed by Jaya. This is also supported by the results shown in Table 3, where the superiority of these two algorithms is obvious since they were the only ones that performed better than the remaining competitors in a statistically significant manner. Statistically significant differences in ranks are denoted with $\dagger$ when the method in the row improves upon the one in the column, and with $\diamond$ when the method in the column improves upon the one in the row. The upper diagonal represents a significance level of $\alpha = 0.9$, and the lower diagonal a level of $\alpha = 0.95$. It is important to note that DE improves even upon Jaya in a statistically significant manner. The performance of DE is not that surprising, considering its effectiveness on a myriad of problems demonstrated in the literature. Yet, the overall performance of Jaya may come as a surprise. Despite being a relatively recent proposal and a comparatively unproven algorithm, it was able to come close to the performance of a well-established one. This suggests its search mechanism to be effective, at least in the case of the problem at hand. The differences amongst the remaining optimisers are not as clear-cut and warrant more of an in-depth look at the performance metrics. When considering the average rankings (FR), two groups can be identified. Inside the worse performing group, consisting of ABC, SGA, PSO, SCA and WOA, no compelling differences are apparent. However, an insight into the potential differences is offered by the provided distance measures, enabling a sub-ordering, where WOA and SCA came out behind the rest. Further, the standard ABC algorithm came out as the strong third contender. Its overall lower performance compared to DE might be attributed to its inherently low convergence-rate due to the search mechanism that updates only a single solution component at a time. Presumably, this may become increasingly notable with a growth in problem dimensionality.

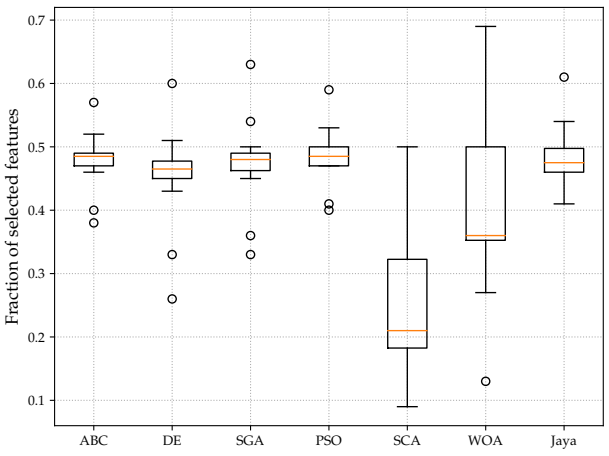

Fig. 1 Box and whisker plots of the average feature subset sizes

Another important aspect of the FS task are certainly the sizes of the selected feature subsets, results of which are reported in Figure 1 in a summarized manner. Presented are the ratios of the number of selected and total features in order to facilitate the interpretation of the results. These are also accompanied by the measure of stability summarized in Figure 2 which is indicative of the stability in finding feature subsets that are consistent across multiple runs. It is called the adjusted stability measure (ASM) [31] and was selected due to its suitability for comparing feature sub-

sets of varying sizes. None of the algorithms included in the comparison is overwhelmingly stable, keeping in mind that the range of ASM lies in $[-1, 1]$. This is due to their stochastic nature and the fact that FS is a multimodal problem where different feature subsets can result in a virtually equal classification performance. Nevertheless, they cannot be deemed unstable for that matter. A different perspective on algorithm stability is offered by the average sizes of the attained subsets. These results suggest that the competing algorithms are relatively consistent, as the fraction of selected features mostly revolves around 50%, apart from SCA and WOA which exhibit a high degree of variability and a contrasting behaviour. When considering the found subset sizes, it can be seen that the overall largest reductions were achieved by the two worst performing algorithms in terms of classification performance (especially SCA). All other algorithms found subsets of similar average sizes. Although not shown here, the observed standard deviation in found subset size for individual datasets was notable and hinted at instability. The obtained feature subsets of varying sizes are in accordance with the aforementioned problem multimodality. The incorporation of the subset size into the objective function is a way of combating this issue, as was stated earlier.

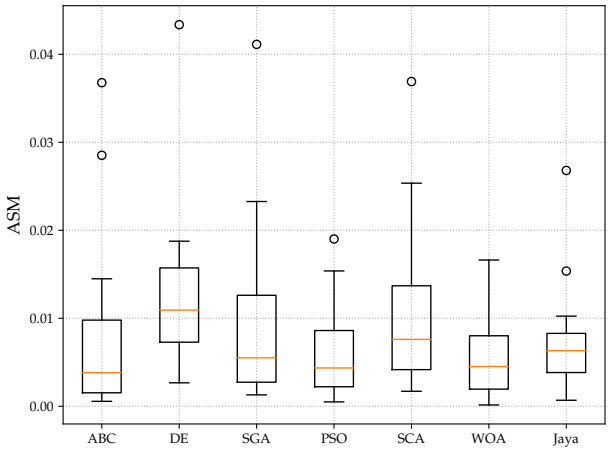

Fig. 2 Box and whisker plots of the average run-to-run stability in terms of obtained feature subsets

Algorithm behaviour during the search process is investigated by considering average rankings during search, obtained by applying the Friedman test across all datasets. These results, recorded in 14 points during the search, are presented in Fig. 3. Notably, Jaya had an overall greater convergence-rate than the others in the beginning of the search. It can be assumed that this is due to its solution update mechanism that moves solutions towards the current best and away from the worst. However, as is visible from the overall average rankings that it was unable to sustain this momentum and DE was able to take the lead shortly after and hold it until the end. The fact that DE achieved the best rankings overall both on the validation and testing data is suggestive of the lack of overfitting. Accordingly, an excessive exploitation might be a reasonable approach only in the case of a restricted number of function evaluations and/or low dimensional datasets. A similar behaviour can also be achieved in DE by incorporating a mutation operator that is focused on exploitation (like best/1 or current-to-best/1). Also, putting a greater emphasis on exploitation in PSO by increasing the influence of social component through parameter configurations (can be achieved by appropriately setting $c_1 < c_2$) could accomplish an analogous effect. Nevertheless, a proper exploration ability seems to be of considerable importance for obtaining good solutions. Achieving a similar effect might be, however, difficult with the more recent of the considered algorithms since their search mechanisms are "fixed" with only a single user-defined parameter being exposed (the population size). Further, Figure 3 also provides support for the earlier statement about the slow convergence of ABC. In the aforementioned figure, WOA stands out as the worst performer in the end, which might be due to its peculiar solution update mechanism, where solutions gradually closer to the best-so-far solution are generated until the whole population converges in the end. It was also the only one amongst the considered algorithms that exhibited spatial convergence of the population, although it does not incorporate a selection procedure and new solutions are accepted regardless of quality. This probably plays a notable role in the competitively low performance it attained since exploration is impaired in the later phases of the search. In turn, this might lead to a waste of function evaluations (the computationally most expensive part of the search) as many similar solutions are created that are likely to result in identical binary vectors. An increase of the population size might represent a remedy in that regard. To be fair, the other algorithms that operate in the real domain are not immune to a possible waste of evaluations, especially in the late phases of the search as the population converges. It should noted that the above-mentioned is an assumption and certainly warrants investigation.

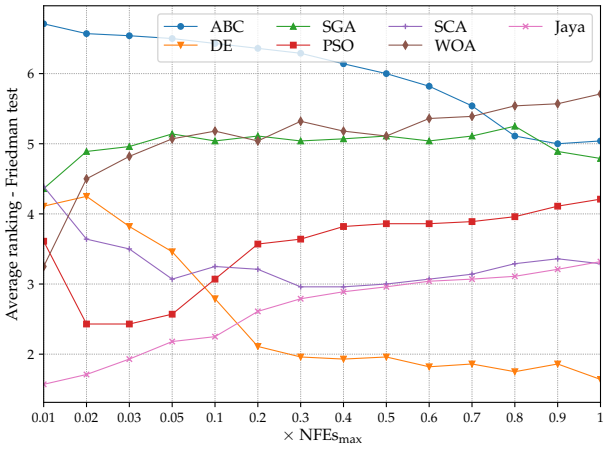

Fig. 3 Average rankings from the Friedman test during search

It is interesting to observe that the only algorithm operating directly with binary solutions, namely SGA, was outperformed by several others that operate on real-valued solutions and later transform them into binary space. Also

interesting is that over half of the considered algorithms utilise solution updating mechanisms that incorporate a movement towards the best-so-far solution. However, as is apparent from the reported results, not all are made equally. The one utilised in Jaya seems to be more effective since it performed notably better than the others. In the end, the performance of PSO left much to be desired, especially since it has been used as the foundation for numerous FS approaches (see, e.g., [5]). Due to such seemingly low performance exhibited by some of the optimisers, post-hoc experiments were conducted in an attempt to see if notable performance improvements can be attained in a simple manner. In particular this refers to the SGA, PSO, WOA and SCA algorithms. In the case of SGA, the survivor selection procedure was changed to incorporate elitism since it is well-known that without it its performance suffers greatly. More precisely, this was approached by incorporating the $\mu + \lambda$ survival selection procedure (with $\mu = \lambda = N$). As for the remaining algorithms, PSO, WOA and SCA, trivial parameter tuning was employed to this end. Accordingly, each of these algorithms were tested with several different parameter configurations. This should also shed some light on the possible limitations discussed earlier regarding the number of user-defined parameters on algorithm flexibility. The results for individual algorithms in the latter group, along with the parameters used, are reported in Table 4. It should be noted that the various parameter configurations for PSO have been taken from the literature, whereas the ones for SCA and WOA have been chosen to cover a reasonable range of population sizes.

Table 4 Overall results in terms of measures derived from the average F1-scores whilst tuning PSO, SCA and WOA

PSO					
Parameter configuration	FR	d_2	d_∞	#Best	#Worst
$c1 = c2 = 1.496$, $\omega = 0.7298$	2.29	0.81	0.42	3	7
$c1 = 2.0412$, $c2 = 0.9477$, $\omega = 0.729$	**1.50**	**0.79**	0.42	**7**	**0**
$c1 = 0.95$, $c2 = 2.85$, $\omega = 0.1$	2.21	0.81	**0.41**	4	7
SCA					
Parameter configuration	FR	d_2	d_∞	#Best	#Worst
$n = 20$	2.89	0.82	0.41	2	1
$n = 30$	3.64	0.83	0.41	2	7
$n = 50$	3.36	0.82	0.41	2	4
$n = 70$	**2.36**	**0.81**	0.41	3	**0**
$n = 100$	2.75	0.81	0.41	**5**	2
WOA					
Parameter configuration	FR	d_2	d_∞	#Best	#Worst
$n = 20$	3.57	0.81	0.41	1	5
$n = 30$	3.57	0.82	0.41	0	3
$n = 50$	2.79	0.80	0.41	3	2
$n = 70$	**2.07**	**0.78**	0.41	**6**	**1**
$n = 100$	3.00	0.82	0.41	4	3

As may be observed, an increase in exploration ability resulted in higher overall performance in the case of all three algorithms. This is suggested by the following. A larger emphasis on the cognitive component (achieved by $c_1 > c_2$) in PSO and an increased population size in SCA and WOA resulted in performance boosts. Surprisingly, WOA was considerably more susceptible to parameter tuning than SCA although both only expose the population size as a tunable parameter. This could be taken as SCA being more robust to varying values of the parameter, however, noting its overall low performance this cannot be accepted as an advantage. It is interesting to note that both performed slightly better for $n = 70$ than for $n = 100$, which can raise the question of further performance improvements when increasing the population size but warrants an additional and more comprehensive investigation for deriving concrete conclusions. As it goes for PSO, the population size of $NS = 30$ was kept fixed throughout all experiments since that value is most commonly used and considered as standard. Certainly, more extensive tuning, at least of the parameters excluding the population size, is necessary in order to paint a broader picture of its true performance potential. Also, changing the neighbourhood structure (or population topology) [32] might be considered (like the ring structure to enhance exploration) as an effective means to boost performance.

In order to properly evaluate the improvements of the algorithms included in the post-hoc experiments and in order to put them into perspective, their best performing variants (suffixed with a $\star$) were compared to the remaining and untuned algorithms (ABC, DE and Jaya). The results of that comparison are shown in Table 5 and 6. Groupings apparent from the original experiment failed to hold i.e. the mutual ordering of the algorithms in terms of overall performance changed. The incorporation of a different and elitist survival selection procedure into SGA pushed it (denoted as $\text{SGA}_{\mu+\lambda}$) near DE performance levels and a group consisting of these two stands out. This could also be discerned from the ranks of the Wilcoxon test where these two algorithms were able to outperform the competition in a statistically significant manner. The other apparent group consists of ABC, PSO, Jaya and, interestingly, WOA which was able to climb up in the rankings. This confirms the previous presumption that an increase of the population size in WOA might yield a performance boost. Despite tuning attempts, SCA was left singled out at the rear in terms of performance (as indicated by all the considered performance measures).

Table 5 Comparison of all algorithms after performance improvement attempts

Alg.	FR	d_2	d_∞	#Best	#Worst
ABC	4.79	0.79	0.42	3	5
DE	**2.57**	**0.76**	**0.40**	2	**0**
$\text{SGA}_{\mu+\lambda}$	2.79	0.77	**0.40**	**5**	2
PSO*	4.64	0.79	0.42	1	2
SCA*	5.21	0.81	0.41	0	4
WOA*	4.21	0.78	0.41	2	1
Jaya	3.79	0.78	0.41	1	**0**

Table 6 Results of the Wilcoxon test for multiple comparisons

	ABC	DE	$\text{SGA}_{\mu+\lambda}$	PSO*	SCA*	WOA*	Jaya
ABC	—	18.0°	23.5°	44.0	60.0	44.0	22.5°
DE	87.0†	—	54.0	89.0†	92.0†	75.0	91.0†
$\text{SGA}_{\mu+\lambda}$	81.5	51.0	—	85.0†	92.0†	74.0	81.5†
PSO*	61.0	16.0°	20.0°	—	63.0	53.0	24.0°
SCA*	45.0	13.0°	13.0°	42.0	—	10.0°	27.0
WOA*	61.0	30.0	31.0	52.0	95.0†	—	45.0
Jaya	82.5	14.0°	23.5	81.0	78.0	60.0	—

5. CONCLUSION

In this paper the performance of several bio-inspired computation algorithms was explored for the feature selection (FS) problem. Several well-known optimisers along

with some more recent ones were selected and experimentally evaluated on diverse datasets. Based on the obtained results the differential evolution (DE) algorithm can be recommended for tackling the FS task. The feature subsets it attained resulted in the overall highest classification performance. Another reasonable suggestion is the genetic algorithm (GA). The simple GA (SGA) did not perform nearly as good as DE, however, further examination revealed promising results attained by a simple replacement of the survivor selection procedure. The main benefit of the GA is that it directly operates in the binary solution space and thus no transformation of candidate solutions is necessary. Further, a favourable aspect of both DE and GA are the numerous operators and enhancements that can be found in the literature. Apart from Jaya, which was able to compete with DE, the other included recent optimisers [sine cosine algorithm (SCA) and whale optimisation algorithm] were seemingly not able to do so. Although some improvements were gained through simple parameter tuning, these were not convincing enough to earn a recommendation.

Run-to-run consistency of found subsets is rarely considered when bio-inspired optimisers are applied to FS but it is something that should not neglected. It is crucial for the interpretability of feature subsets as well as for providing insight into feature interactions and relevance. Accordingly, appropriate measures should be taken as to ensure relative stability when developing algorithms for FS. Various approaches that can be considered for that purpose are presented in [33]. This is where DE again performed generally better than the other competing optimisers, albeit neither can be regarded particularly stable. Another possible direction for future work could be a more extensive analysis including more bio-inspired optimisers as well as larger datasets, both in terms of sample and feature count. The latter represents a more serious issue and should be kept in mind since, according to the literature (see, e.g., [5]), these approaches are not suitable when dealing with several thousand features. Pre-processing steps, like an initial reduction via filter-based approaches or clustering, could provide aid in this case and warrant consideration. Further, the employed parameter values were taken as recommended by the literature and represent the approach likely to be taken by an aspiring practitioner. However, performance gains are to be had with parameter tuning as is demonstrated by the additional conducted experiments. Yet, the extent of these improvements could prove to be inconsequential for some algorithms as was the case with SCA.

Finally, this paper might be considered as another critique on the metaphor-centric algorithm development approach. Hence, it should be remarked that this was not the intention. However, the presented results do not provide any compelling reasons for favouring recent proposals over the much older algorithms and question their significance to the bio-inspired computation community. In the end, the well-established algorithms proved be the reasonable choice for developing FS approaches.

ACKNOWLEDGEMENT

This work was supported by the European Regional Development Fund under the grants KK.01.2.1.01.0127, and KK.01.1.1.01.0009 (DATACROSS).

REFERENCES

[1] CHANDRASHEKAR, G. – SAHIN, F.: A survey on feature selection methods. Computers and Electrical Engineering, Vol. 40, No. 1, 16–28, 2014.

[2] CAI, J. – LUO, J. – WANG, S. – YANG, S.: Feature selection in machine learning: A new perspective. Neurocomputing, Vol. 300, 70–79, 2018.

[3] JOVIĆ, A. – BRKIĆ, K. – BOGUNOVIĆ, N.: A review of feature selection methods with applications. In: 2015 International Convention on Information, Communication and Electronic Technology (MIPRO), Opatija, pp.1200–1205, 2015.

[4] REMESEIRO, B. – BOLON-CANEDO, V.: A review of feature selection methods in medical applications. Computers in Biology and Medicine, Vol. 112, pp. 103375, 2019.

[5] XUE, B. – ZHANG, M. – BROWNE, W. N. – YAO, X.: A Survey on Evolutionary Computation Approaches to Feature Selection. IEEE Transactions on Evolutionary Computation, Vol. 20, No. 4, pp. 606–626, 2016.

[6] DEL SER, J. ET AL.: Bio-inspired computation: Where we stand and what's next. Swarm and Evolutionary Computation, Vol. 48, pp. 220–250, 2019.

[7] YU, L. – LIU, H.: Efficient Feature Selection via Analysis of Relevance and Redundancy. Journal of Machine Learning Research, Vol. 5, pp. 1205–1224, 2004.

[8] GUYON. I – ELISSEEFF, A.: An Introduction to Variable and Feature Selection. Journal of Machine Learning Research, Vol. 3, pp. 1157–1182, 2003.

[9] STAŃCZYK, U.: Feature evaluation by filter, wrapper, and embedded approaches. In: Feature Selection for Data and Pattern Recognition, Stańczyk, U. and Jain, L. C. (eds.), pp.29-44, 2015.

[10] EIBEN, A. E. – SMITH, J. E.: Introduction to evolutionary computing. Springer-Verlag Berlin Heidelberg, 2015.

[11] SHI, Y. – EBERHART, R.: A modified particle swarm optimizer. In: 1998 IEEE International Conference on Evolutionary Computation (ICEC), Anchorage, pp. 69–73, 1998.

[12] STORN, R. – PRICE, K.: Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, Vol. 11, No. 4, pp. 341–359, 1997.

[13] RAJPUROHIT, J. – SHARMA, T. K. – AJITH, A. – VAISHALI: Glossary of Metaheuristic Algorithms. International Journal of Computer Information Systems and Industrial Management Applications, Vol. 9, pp. 181–205, 2017.

[14] SÖRENSEN, K. – SEVAUX, M. – Glover, F.: A History of Metaheuristics. In: Handbook of Heuristics, Martí, R. – Pardalos, P. – Resende, M. (eds.), 2018.

[15] LONES, M. A.: Metaheuristics in Nature-inspired Algorithms. In: Genetic and Evolutionary Computation Conference (GECCO), Vancouver, pp. 1419–1422, 2014.

[16] SÖRENSEN, K.: Metaheuristics—the metaphor exposed. International Transactions in Operational Research, Vol. 22, No. 1, pp. 1–38, 2015.

[17] MIRJALILI, S.: SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-Based Systems, Vol. 96, pp. 120–1333, 2016.

[18] MIRJALILI, S. – LEWIS, A.: The Whale Optimization Algorithm. Advances in Engineering Software, Vol. 95, pp. 51–67, 2016.

[19] VENKATA RAO, R.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, Vol. 7, No. 1, pp. 19–34, 2016.

[20] MARTINOVIĆ, G – BAJER, D. – ZORIĆ, B.: A differential evolution approach to dimensionality reduction for classification needs. International Journal of Applied Mathematics and Computer Science, Vol. 24, No. 1, pp. 111–122, 2014.

[21] EMARY, E. – ZAWBAA, H. M. – HASSANIEN, A. E.: Binary ant lion approaches for feature selection. Neurocomputing, Vol. 213, pp. 54–65, 2016.

[22] ARORA, S. – ANAND, P.: Binary butterfly optimization approaches for feature selection. Expert Systems with Applications, Vol. 116, pp. 147–160, 2019.

[23] LÓPEZ, V. ET AL.: An Insight Into Classification With Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics. Information Sciences, Vol. 250, pp. 113–141, 2013.

[24] JAPKOWICZ, N. – SHAH, M.: Evaluating learning algorithms: a classification perspective. Cambridge University Press, 2011.

[25] BACHE, K. – LICHMAN, M.: UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, `http://archive.ics.uci.edu/ml`

[26] ALCALÁ-FDEZ, J. ET AL.: KEEL Data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Journal of Multiple-Valued Logic and Soft Computing, Vol. 17, No. 2–3, pp. 255–287, 2011.

[27] KARABOGA, D. – Basturk, B.: A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm. Journal of Global Optimization, Vol. 39, No. 3, pp. 459–471, 2007.

[28] MERNIK, M. ET AL.: On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation. Information Sciences, Vol. 291, pp. 115–127, 2015.

[29] ZORIĆ, B. – BAJER, D. – MARTINOVIĆ, G.: Utilising Filter Inferred Information in Nature-inspired Hybrid Feature Selection. In: 2018 International Conference on Smart Systems and Technologies (SST), pp. 117–123, 2018.

[30] BAJER, D. – ZORIĆ, B. – DUDJAK, M. – MARTINOVIĆ, G.: Performance Analysis of SMOTE-Based Oversampling Techniques When Dealing with Data Imbalance. In: 2019 International Conference on Systems, Signals and Image Processing, Osijek, pp. 265–271, 2019.

[31] LUSTGARTEN, J. L. – GOPALAKRISHNAN, V. – VISWESWARAN, S.: Measuring stability of feature selection in biomedical datasets. In: 2009 AMIA annual symposium, San Francisco, pp. 406, 2009.

[32] MUÑOZ ZAVALA, A. E.: A Comparison Study of PSO Neighborhoods. In: EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II, Schütze, O. et al. (eds.), pp. 251–265, 2013.

[33] KHAIRE, U. M. – DHANALAKSHMI, R.: Stability of feature selection algorithm: A review. Journal of King Saud University-Computer and Information Sciences, Vol. in press, 2019.

Received March 2, 2020, accepted June 1, 2020

BIOGRAPHIES

Dražen Bajer received his Bachelor, Master and Ph.D. degrees in computer engineering from the Faculty of Electrical Engineering, Computer Science and Information Technology, J.J. Strossmayer University of Osijek in 2008, 2010, and 2017 respectively. His research interests include supervised computational intelligence methods and their applications, and (un)supervised classification.

Bruno Zorić received his Bachelor, Master and Ph.D. degrees in computer engineering from the Faculty of Electrical Engineering, Computer Science and Information Technology, J.J. Strossmayer University of Osijek in 2008, 2011, and 2017 respectively. His research interests include supervised classification, computational intelligence and affective computing.

Mario Dudjak received his Bachelor and Master degrees in computer engineering from the Faculty of Electrical Engineering, Computer Science and Information Technology, J.J. Strossmayer University of Osijek in 2016 and 2018, respectively. He is currently pursuing his Ph.D. degree at the same institution. His research interests are machine learning and supervised classification.

Goran Martinović is a full professor of computer science. He obtained his B.Sc.E.E. degree from the Faculty of Electrical Engineering, J.J. Strossmayer University of Osijek in 1996. In 2000 and 2004, he obtained his M.Sc. and Ph.D. degrees in computer science, both from the Faculty of Electrical Engineering and Computing, University of Zagreb. His research interests include computational intelligence, data analysis, distributed computer systems and real-time systems. He is a member of the IEEE, ACM and KOREMA and IEEE SMC Technical Committee on Distributed Intelligent Systems.