

Pruning Optimum-Path Forest Ensembles Using Quaternion-based Optimization

Silas Evandro Nachif Fernandes
Department of Computing,
Federal University of São Carlos,
Rod. Washington Luís, Km 235,
São Carlos, SP, 13565-905, Brazil
Email: silas.fernandes@dc.ufscar.br

João Paulo Papa
Department of Computing,
São Paulo State University,
Av. Eng. Luiz Edmundo Carrijo Coube, 14-01,
Bauru, SP, 17033-360, Brazil.
Email: papa@fc.unesp.br

Abstract—Machine learning techniques have been actively pursued in the last years, mainly due to the great number of applications that make use of some sort of intelligent mechanism for decision-making processes. In this context, we shall highlight pruning strategies, which provide heuristics to select from a collection of classifiers the ones that can really improve recognition rates when working together. In this paper, we present an ensemble pruning approach of Optimum-Path Forest classifiers based on metaheuristics, as well as we introduced the concept of quaternions in ensemble pruning strategies. Experimental results over synthetic and real datasets showed the effectiveness and efficiency of the proposed approach for classification problems.

I. INTRODUCTION

The use of multiple classifiers has become an area of great interest in pattern recognition, being highly motivated due to the likeliness in including other classifiers during the decision process as a good approach to improve the generalization ability of the ensemble. Also, the strategy of polling the decisions of classifiers that are complementary to each other can improve the classification accuracy of an ensemble with respect to the individual learners.

Different methods to combine classifiers have been proposed, such as voting [1], weighted voting [2] and methods based on Dempster-Shafer theory of evidence [3]. Some authors model the combination of classifiers as an optimization problem [4], where the weights of the confidence levels for each classifier are determined by evolutionary optimization techniques [5].

However, combining a large number of classifiers requires a considerable memory at the price of a slower classification phase. One way to speed up such process is the selection of a subset of the classifiers from the original ensemble. Such approach, also known as *ensemble pruning*, has provided several benefits in different approaches [6], [7]. Usually, large ensembles of classifiers may comprise both high and low predictive performance models [8]. Pruning these low predictive models for a given classification problem while maintaining the high predictive ones may contribute to improve the overall predictive performance of the whole ensemble. In general, ensemble pruning selects a subset of the models, thus providing efficiency by reducing the ensemble size prior to combine the model.

It is worth noting to highlight that finding the optimal sub-ensemble is a hard problem, whose solution is computationally expensive [9]. Therefore, an interesting way of dealing with such problem is to model ensemble pruning as a metaheuristic-driven optimization task, such as by means of Genetic Programming [6]. In general, is expected that Genetic Programming may provide a near-optimal ensemble selection with a significant reduction of the number of classifiers. A number of works also employed metaheuristics to prune ensembles of classifiers as well [10], [11], [12], with very suitable results.

Some years ago, Papa et al. [13], [14], [15] introduced the Optimum-Path Forest (OPF) classifier, which is a graph-based pattern recognition technique that uses a generalization of Dijkstra algorithm for multiple sources and path-cost functions. The OPF classifier has demonstrated interesting results in terms of efficiency and effectiveness, being some of them comparable to the ones obtained by Support Vector Machines, but usually faster for training, since OPF [13], [14] is parameterless and its training step takes $\theta(m^2)$, where m stands for the training set size.

However, the reader can refer to a very few works that deal with the problem of combining ensembles of OPF classifiers. Ponti and Papa [16], for instance, showed the training step of the OPF classifier can be more efficient and effective when using combination of disjoint training subsets. Furthermore, Ponti et al. [17] proposed the combination of OPF classifiers using Markov Random Fields as a decision graph, as well as Game Theory to compute the final decision, i.e., each classifier is seen as player and each classifier decision (class label) is seen as a strategy. Moreover, to the best of our knowledge, we have not observed any work that attempted at pruning ensembles of OPF-based classifiers.

Therefore, the main contributions of this work are twofold: (i) to investigate the use of ensemble pruning methods based on metaheuristic techniques considering the OPF classifier, as well as (ii) to evaluate the effectiveness and efficiency of ensemble pruning when modeled as an optimization task in a quaternion-driven space. In this work, we evaluated five different optimization algorithms for ensemble pruning, say that Particle Swarm Optimization (PSO) [18], Harmony Search

(HS) [19], Quaternion-based Harmony Search (QHS) [20], Cuckoo Search (CS) [21] and Firefly Algorithm (FFA) [22].

The remainder of the paper is organized as follows. Sections II and III present the OPF and quaternion-based representation background theories, respectively. Section IV discusses the proposed approach for ensemble pruning, and Section V describes the methodology and the experimental results. Finally, conclusions and future works are stated in Section VI.

II. OPTIMUM-PATH FOREST

In this section, the theoretical foundation of the naïve OPF is briefly presented. Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$ be a λ -labeled dataset, such that \mathcal{D}_1 and \mathcal{D}_2 stand for the training and test sets, respectively. Let $\mathcal{S} \subset \mathcal{D}_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes). Let (\mathcal{D}_1, A) be a complete graph whose nodes are the samples in \mathcal{D}_1 , and any pair of samples defines an arc in $A = \mathcal{D}_1 \times \mathcal{D}_1$. Additionally, let π_s be a path (i.e. a sequence of adjacent and distinct samples) in (\mathcal{D}_1, A) with terminus at sample $s \in \mathcal{D}_1$. The OPF algorithm employs the path-cost function f_{\max} , as follows:

$$\begin{aligned} f_{\max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in \mathcal{S} \\ +\infty & \text{otherwise,} \end{cases} \\ f_{\max}(\pi_s \cdot \langle s, t \rangle) &= \max\{f_{\max}(\pi_s), d(s, t)\}, \end{aligned} \quad (1)$$

where $d(s, t)$ stands for the distance between nodes s and t . Therefore, $f_{\max}(\pi_s)$ computes the maximum distance among adjacent samples in π_s , when π_s is not a trivial path. In short, the OPF algorithm tries to minimize $f_{\max}(\pi_t), \forall t \in \mathcal{D}_1$.

In regard to the *training phase*, the optimum set of prototypes, say that S^* , can be found by exploiting the theoretical relation between the minimum-spanning tree and the optimum-path tree for f_{\max} [23]. Essentially, the training step consists in finding both S^* and an OPF classifier rooted at S^* . By computing a minimum spanning tree (MST) in the complete graph (\mathcal{D}_1, A) , one obtain a connected acyclic graph whose nodes are all samples of \mathcal{D}_1 , and the arcs are undirected and weighted by the distances d among adjacent samples. After that, a competition process among samples in \mathcal{S} takes place, which try to conquer samples in $\bar{\mathcal{S}}$ using f_{\max} (Equation 1).

In regard to the *classification phase*, for any sample $t \in \mathcal{D}_2$, we consider all arcs connecting t with samples $s \in \mathcal{D}_1$, as though t were part of the training graph. Considering all possible paths from S^* to t , we find the optimum path $P^*(t)$ from S^* and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. This path can be identified incrementally, by evaluating the optimum cost $C(t)$ as follows:

$$C(t) = \min_{\forall s \in \mathcal{D}_1} \{\max\{C(s), d(s, t)\}\}. \quad (2)$$

Let the node $s^* \in \mathcal{D}_1$ be the one that satisfies Equation 2 (i.e., the predecessor $P(t)$ in the optimum path $P^*(t)$). Given that $L(s^*) = \lambda(R(t))$, the classification simply assigns $L(s^*)$ as the class of t . An error occurs when $L(s^*) \neq \lambda(t)$.

III. QUATERNION ALGEBRA

A *quaternion* q is composed of real and complex numbers, i.e., $q = x_0 + x_1i + x_2j + x_3k$, where $x_0, x_1, x_2, x_3 \in \mathbb{R}$ and i, j, k are imaginary numbers following the next set of equations:

$$ij = k, \quad jk = i, \quad ki = j, \quad ji = -k, \quad kj = -i, \quad (3)$$

and

$$ik = -j, \quad i^2 = j^2 = k^2 = 1. \quad (4)$$

Roughly speaking, a quaternion q is represented in a 4-dimensional space over the real numbers, i.e., \mathbb{R}^4 . Actually, we can consider the real numbers only, since most applications do not consider the imaginary part, as the one addressed in this work.

Given two quaternions $q_1 = x_0 + x_1i + x_2j + x_3k$ and $q_2 = y_0 + y_1i + y_2j + y_3k$, the quaternion algebra defines a set of main operations [24]. The addition, for instance, can be defined by:

$$\begin{aligned} q_1 + q_2 &= (x_0 + x_1i + x_2j + x_3k) + (y_0 + y_1i + y_2j + y_3k) \\ &= (x_0 + y_0) + (x_1 + y_1)i + (x_2 + y_2)j + (x_3 + y_3)k, \end{aligned} \quad (5)$$

while the subtraction is defined as follows:

$$\begin{aligned} q_1 - q_2 &= (x_0 + x_1i + x_2j + x_3k) - (y_0 + y_1i + y_2j + y_3k) \\ &= (x_0 - y_0) + (x_1 - y_1)i + (x_2 - y_2)j + (x_3 - y_3)k. \end{aligned} \quad (6)$$

Another important operation is the norm, which maps a given quaternion to a real-valued number, as follows:

$$\begin{aligned} N(q_1) &= N(x_0 + x_1i + x_2j + x_3k) \\ &= \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2}. \end{aligned} \quad (7)$$

Finally, Fister et al. [25] introduced two other operations, *grand* and *qzero*. The former initializes a given quaternion with values drawn from a Gaussian distribution, and it can be defined as follows:

$$grand() = \{x_i = \mathcal{N}(0, 1) | i \in \{0, 1, 2, 3\}\}. \quad (8)$$

The latter function initialized a quaternion with zero values, as follows:

$$qzero() = \{x_i = 0 | i \in \{0, 1, 2, 3\}\}. \quad (9)$$

Although there are other operations, we defined only the ones employed in this work.

IV. ENSEMBLE PRUNING AS A METAHEURISTIC-BASED OPTIMIZATION

In this section, we describe the proposed approach to prune ensembles based on metaheuristic techniques. The OPF classifier uses the *abstract output* method when classifying samples, i.e., the output of the classifier is a single label. Xu et al. [26] defined an interesting approach to combine the outputs of L classifiers in an ensemble depending on the information obtained from the individual members. Such approach considers that each classifier assigns a class label to every sample in the dataset. Therefore, the ensemble of classifiers generates a collection of L possible outputs to each sample to be classified.

Consider an ensemble of L OPFs in a bagging approach, i.e. the classifiers are aggregated by using different bootstrapped samples of the original training data. Let $\mathcal{M} = \{M_1, M_2, \dots, M_L\}$ be a set of L classifiers, and $\Omega = \{\omega_1, \omega_2, \dots, \omega_K\}$ be a set of K class labels. Each classifier takes an n -dimensional input vector and associates it to a class label, i.e., $M_i : \mathbb{R}^n \rightarrow \Omega, i = 1, 2, \dots, L$. Therefore, for any sample to be classified, the ensemble of classifiers generates a collection $\Psi_z = [\psi_z(M_1), \dots, \psi_z(M_L)]$ of possible outputs, where $\psi_z(M_i)$ stands for the output of classifier M_i considering sample z .

However, instead of considering the outputs of all classifiers, the idea of ensemble pruning is to select a subset $\mathcal{M}' \subset \mathcal{M}$ such that the recognition rate over a validating set is maximized. Let $\mathcal{D}_v \subset \mathcal{D}_1$ be a validating set, and $\hat{\mathcal{D}}_1 = \mathcal{D}_1 \setminus \mathcal{D}_v$ be an L -folded training set such that $\hat{\mathcal{D}}_1 = \hat{\mathcal{D}}_1^1 \cup \hat{\mathcal{D}}_1^2 \cup \dots \cup \hat{\mathcal{D}}_1^L$. The main idea is to train each classifier M'_i over $\hat{\mathcal{D}}_1^i$, and then classify \mathcal{D}_v using the majority voting. In short, the idea is to turn “on” or “off” each classifier in \mathcal{M} to build \mathcal{M}' , and then classify \mathcal{D}_v . The task to consider or not a given classifier is performed by the metaheuristic optimization technique, which essentially aims at learning what classifier will be turned “on” or “off”. Therefore, such choice will be guided by the maximum accuracy over \mathcal{D}_v .

In order to decide whether we are going to select a given classifier or not, we associate a weight $w_i \in [0, 1]$ to each classifier, which is further selected to compose \mathcal{M}' if $w_i > \tau$, being τ an adaptive threshold [27] updated as follows:

$$\tau = \rho - \sigma, \quad (10)$$

where ρ is the mean weight, and σ is calculated by:

$$\sigma = \sqrt{\frac{1}{m} \sum_{i=1}^m (w_i - \rho)^2} \quad \forall w_i < \rho, \quad (11)$$

where m is the number of classifiers whose weight is smaller than ρ . Notice the above equation considers a classifier M_i whose weight w_i is smaller than ρ only.

In short, the main idea is to model each possible solution of the search space as a set of weights, and the optimization technique problem aims at finding the best values for such weights that maximize the OPF accuracy over \mathcal{D}_v . Figure 1

illustrates the above situation, and Algorithm 1 implements the proposed approach for ensemble pruning.

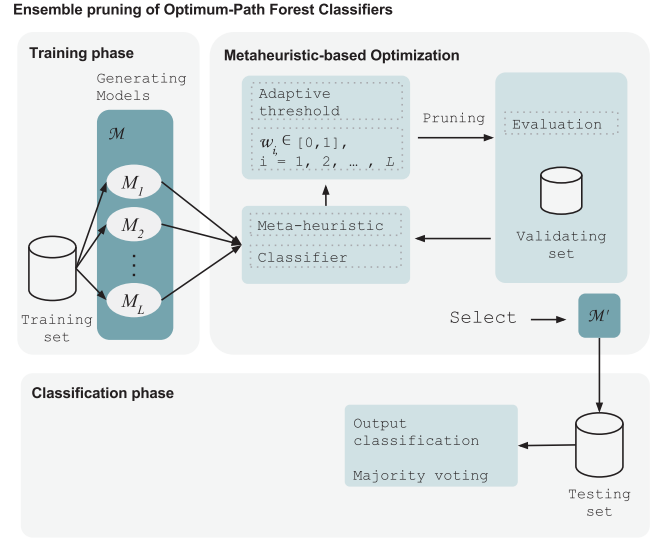


Fig. 1. Ensemble pruning process.

Algorithm 1: Ensemble pruning

Input: A labeled training set \mathcal{D}_1 , a labeled validating set \mathcal{D}_v , and a set \mathcal{M} of L OPF classifiers.
Output: A subset \mathcal{M}' of classifiers.

- 1 /* Training phase */
- 2 **for** $i = 1$ to L **do**
- 3 Take a bootstrap dataset \mathcal{D}_1^i from \mathcal{D}_1 ;
- 4 Build a classifier \mathcal{M}_i using the training set \mathcal{D}_1^i ;
- 5 /* Metaheuristic-based optimization process */
- 6 **for each iteration do**
- 7 $\mathcal{W} = [w_1, w_2, \dots, w_L] \leftarrow$ Perform Quaternion-based Optimization;
- 8 **for** $i = 1$ to L **do**
- 9 **if** $w_i > \tau$ **then**
- 10 $\mathcal{M}' \leftarrow \mathcal{M}_i$;
- 11 $\Psi(\mathcal{M}') \leftarrow$ Classify \mathcal{D}_v using \mathcal{M}' ;
- 12 $accuracy \leftarrow$ Majority voting of $\Psi(\mathcal{M}')$ over \mathcal{D}_v ;
- 13 **return** \mathcal{M}'

A. Optimizing with Quaternions

Usually, it is expected that optimization with quaternions may produce smoother fitness landscapes, thus designing surfaces with less local optima [25], [20]. The idea is to model each real-valued decision variable to be optimized (i.e. each classifier’s weight in the ensemble) as being another real-valued variable, but now in \mathbb{R}^4 , since a quaternion comprises four real variables, as discussed in Section III. Therefore, we have a *tensor* $T_{4 \times L}$ in the optimization space for each possible

solution, since we have a 4-dimensional quaternion for each one of the L classifiers.

We use the norm of each quaternion (Equation 7) to map it to a real-valued variable aiming at performing standard optimization. However, it is important to highlight we play with the values in the quaternion space, for further mapping them to standard Euclidean space. This means we are interested in finding good representations in the quaternion space first. Figure 2 illustrates the above situation.

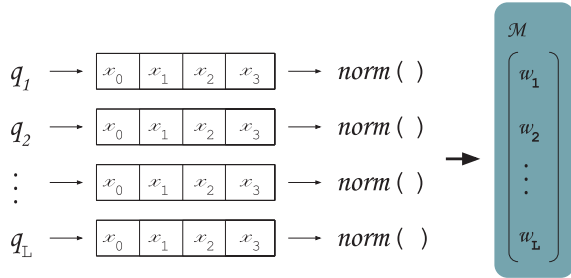


Fig. 2. Quaternion-based encoding process.

V. METHODOLOGY AND EXPERIMENTAL RESULTS

The proposed metaheuristic-based ensemble pruning approach concerning OPF classifiers is compared against standard OPF over 6 real and synthetic benchmark classification datasets¹²³, whose main characteristics are presented in Table I⁴.

TABLE I
DESCRIPTION OF THE DATASETS.

Dataset	# samples	# features	# classes
Pima-Indians-Diabetes	768	8	2
Statlog-Australian	690	14	2
Statlog-Heart	270	13	2
Synthetic1	500	2	2
Synthetic2	1,000	2	2
UCI-Breast-Cancer	683	10	2

In order to obtain statistically meaningful results, three different ranges of training, validating and testing data were used: (i) in a first stage, each dataset was partitioned into three subsets, say that training (50%), initial validating (10%) and testing sets (40%), hereinafter denoted as 50:10:40; (ii) in a second stage, the datasets were partitioned as 45:20:35; and (iii) in the last stage, the datasets were partitioned as 40:30:30. For each range, training, validating and testing sets were selected randomly and the process was repeated 20 times (cross-validation)⁵.

¹<http://mldata.org>

²<http://archive.ics.uci.edu/ml>

³<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>

⁴The experiments were conducted on a computer with a Pentium Intel Core i5® 3.2Ghz processor, 4 GB of memory RAM and Linux Ubuntu Desktop LTS 64 bits 14.04.1 as the operational system

⁵Notice the percentages have been empirically chosen, being more intuitive to provide a larger validating set for ensemble pruning learning

In order to allow a fair comparison, we trained standard OPF over $\mathcal{D}_1 \cup \mathcal{D}_v$ considering the aforementioned three stages. In addition, the mean accuracy and computational load (in seconds) are also computed for each technique. Notice the idea of using different percentages for \mathcal{D}_1 , \mathcal{D}_v and \mathcal{D}_2 is motivated to assess the effectiveness of the proposed approach under distinct scenarios. The final results were evaluated through the Wilcoxon signed-rank test with significance of 0.05 [28]. For the sake of implementation purposes, we used the open-source libraries LibOPF⁶ and LibOPT⁷.

In order to evaluate the influence of different ensembles, we performed a comparison with ensembles comprising 3, 5, 7 and 9 classifiers. Furthermore, in regard to the metaheuristic optimization, we opted to employ the following techniques:

- *Harmony Search* (HS): 5 harmonies with 20 iterations, $HMCR = 0.7$, $PAR = 0.7$ and $\beta = 10$. Variables $HMCR$ and PAR , which stand for “Harmony Memory Considering Rate” and “Pitch Adjusting Rate”, are used to guide HS onto the search space, as well as to avoid traps from local optima.
- *Quaternion-based Harmony Search* (QHS): an improved variant of HS based on quaternions [20]. Such variant has the very same input configuration used in HS.
- *Particle Swarm Optimization* (PSO): 5 particles with 20 iterations, $c_1 = 1.4$, $c_2 = 0.6$ and $\xi = 0.7$. Variables c_1 and c_2 are used to weight the importance of a possible solution being far or close to the local and global optimum, respectively. Variable ξ stands for the well-known “inertia weight”, which is used as a step size towards better solutions.
- *Firefly Algorithm* (FFA): population size of 5 with 20 iterations, $\gamma = 0.3$ and $\mu = 1.0$. The variables γ and μ are used to control the randomness and the attractiveness, respectively.
- *Cuckoo Search* (CS): 5 particles with 20 iterations, and $p_a = 0.25$. Variable p_a is used to control the elitism and the balance of the randomization and local search. In this case, few parameters in CS make an algorithm less complex and thus potentially more generic.

Table II presents the mean accuracies and standard deviation over all datasets, being the recognition rates computed according to Papa et al. [13]. The values in bold stand for the most accurate techniques according of the Wilcoxon signed-rank test. We can observe the proposed OPF ensemble pruning has obtained the best results in almost all datasets. Consider “Synthetic2” dataset (Figure 3), which comprises a high level of overlapping among samples from different classes. In this case, ensemble pruning was considerably more effective. Therefore, the use of multiple classifiers may be more effective when applied to more complex problems.

Table III presents the mean F-measure results concerning the very same group of datasets using 9 classifiers. In this case, all the pruning-based approaches shown the similar behavior. The

⁶<https://github.com/jppbsi/LibOPF.git>

⁷<https://github.com/jppbsi/LibOPT.git>

TABLE II
MEAN ACCURACY RESULTS.

Dataset	Approach	Ensemble of 3 OPF	Ensemble of 5 OPF	Ensemble of 7 OPF	Ensemble of 9 OPF
Pima-Indians-Diabetes	OPF _(baseline)	65.50 ± 2.08	65.50 ± 2.08	65.50 ± 2.08	65.50 ± 2.08
	OPF _{CS}	66.87 ± 2.78	68.81 ± 2.46	68.70 ± 3.30	67.65 ± 2.84
	OPF _{FFA}	63.50 ± 6.01	68.14 ± 5.23	66.33 ± 5.96	65.95 ± 6.00
	OPF _{HS}	66.90 ± 2.61	68.73 ± 2.44	68.39 ± 3.23	68.26 ± 2.52
	OPF _{PSO}	67.33 ± 3.39	69.30 ± 2.72	68.09 ± 3.08	67.72 ± 3.41
	OPF _{QHS}	66.90 ± 2.77	67.07 ± 3.21	67.67 ± 2.96	66.88 ± 2.78
Statlog-Australian	OPF _(baseline)	77.96 ± 2.04	77.96 ± 2.04	77.96 ± 2.04	77.96 ± 2.04
	OPF _{CS}	77.02 ± 3.21	80.65 ± 2.52	82.02 ± 1.83	83.57 ± 1.79
	OPF _{FFA}	75.59 ± 6.19	78.38 ± 7.38	81.66 ± 4.67	80.38 ± 7.57
	OPF _{HS}	78.20 ± 3.00	81.71 ± 1.93	82.82 ± 2.30	83.52 ± 2.65
	OPF _{PSO}	76.64 ± 3.06	81.18 ± 2.28	82.40 ± 2.20	83.86 ± 2.16
	OPF _{QHS}	75.94 ± 2.95	81.28 ± 1.65	81.88 ± 2.24	82.39 ± 2.68
Statlog-Heart	OPF _(baseline)	74.94 ± 1.91	74.94 ± 1.91	74.94 ± 1.91	74.94 ± 1.91
	OPF _{CS}	78.78 ± 2.85	77.66 ± 1.63	80.12 ± 3.55	81.10 ± 3.50
	OPF _{FFA}	70.89 ± 6.71	79.48 ± 4.57	80.30 ± 1.59	80.64 ± 3.32
	OPF _{HS}	76.18 ± 2.39	80.91 ± 2.40	80.32 ± 2.81	80.91 ± 2.18
	OPF _{PSO}	73.85 ± 2.99	79.76 ± 2.78	80.05 ± 1.43	81.14 ± 2.28
	OPF _{QHS}	76.97 ± 4.21	77.48 ± 2.73	79.53 ± 3.81	80.25 ± 3.82
Synthetic1	OPF _(baseline)	53.98 ± 2.89	53.98 ± 2.89	53.98 ± 2.89	53.98 ± 2.89
	OPF _{CS}	53.86 ± 3.62	55.84 ± 2.61	55.80 ± 3.40	56.31 ± 3.02
	OPF _{FFA}	54.79 ± 2.55	55.76 ± 3.76	56.04 ± 3.56	54.71 ± 2.93
	OPF _{HS}	52.97 ± 4.37	55.83 ± 3.21	56.33 ± 3.90	55.78 ± 3.22
	OPF _{PSO}	53.38 ± 4.47	54.79 ± 2.60	55.78 ± 4.26	56.29 ± 2.71
	OPF _{QHS}	53.51 ± 3.18	55.52 ± 3.16	55.16 ± 3.79	55.23 ± 3.69
Synthetic2	OPF _(baseline)	71.65 ± 2.05	71.65 ± 2.05	71.65 ± 2.05	71.65 ± 2.05
	OPF _{CS}	71.89 ± 2.89	75.89 ± 2.03	77.25 ± 2.19	77.47 ± 2.60
	OPF _{FFA}	71.04 ± 5.90	75.69 ± 3.61	75.31 ± 5.31	77.39 ± 3.34
	OPF _{HS}	73.29 ± 2.43	76.31 ± 2.37	77.03 ± 1.91	77.87 ± 1.81
	OPF _{PSO}	73.35 ± 2.45	76.33 ± 2.42	77.04 ± 2.09	78.15 ± 1.68
	OPF _{QHS}	71.92 ± 2.50	75.67 ± 2.65	76.56 ± 2.29	76.87 ± 2.37
UCI-Breast-Cancer	OPF _(baseline)	94.42 ± 1.05	94.42 ± 1.05	94.42 ± 1.05	94.42 ± 1.05
	OPF _{CS}	92.33 ± 2.95	94.85 ± 1.63	95.50 ± 1.49	95.57 ± 1.31
	OPF _{FFA}	88.11 ± 14.48	92.18 ± 6.60	93.66 ± 6.29	92.73 ± 6.71
	OPF _{HS}	94.10 ± 2.68	95.42 ± 1.20	96.17 ± 1.19	96.09 ± 0.97
	OPF _{PSO}	91.76 ± 3.36	94.94 ± 1.22	96.23 ± 1.07	95.70 ± 0.87
	OPF _{QHS}	91.14 ± 3.16	94.89 ± 1.87	95.05 ± 2.09	95.55 ± 1.64

values in bold stand for the most accurate techniques according of the Wilcoxon signed-rank test.

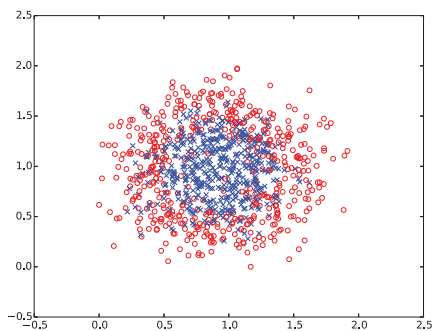


Fig. 3. “Synthetic2” dataset.

In order to provide a robust statistical analysis, we performed the non-parametric Friedman test, which is used to rank the algorithms for each dataset separately. In case of Friedman test provides meaningful results to reject the null-hypothesis (h_0 : all techniques are equivalent), we can perform

a post-hoc test further. For this purpose, we conducted the Nemenyi test, proposed by Nemenyi [29] and described by Demšar [30], which allows us to verify whether there is a critical difference (CD) among techniques or not. The results of the Nemenyi test can be represented in a simple diagram, in which the average ranks of the methods are plotted on the horizontal axis, where the lower the average rank is, the better the technique is. Moreover, the groups with no significant difference are then connected with a horizontal line.

Figure 4 depicts the statistical analysis considering the accuracy results for ensemble pruning using 9 classifiers. As one can observe, the pruning-based approaches can be considered the most accurate ones by Nemenyi test. Such point reflects the OPF ensemble pruning using metaheuristic techniques achieved the best accuracy rates in the majority of datasets. Furthermore, the HS approach can be considered the most accurate technique. However, the statistical test did not point out a CD between HS, PSO and CS metaheuristics in the first group, which means they performed similarly.

Figure 5 depicts the statistical analysis considering the F-Measure results for ensemble pruning using 9 classifiers. In

TABLE III
MEAN F-MEASURE.

Dataset	OPF _(baseline)	OPF _{CS}	OPF _{FFA}	OPF _{HS}	OPF _{PSO}	OPF _{QHS}
Pima-Indians-Diabetes	0.5371 ± 0.04	0.5812 ± 0.04	0.5833 ± 0.04	0.5942 ± 0.04	0.5712 ± 0.05	0.5978 ± 0.04
Statlog-Australian	0.8087 ± 0.02	0.8603 ± 0.02	0.8608 ± 0.02	0.8617 ± 0.01	0.8616 ± 0.01	0.8578 ± 0.02
Statlog-Heart	0.7350 ± 0.05	0.7939 ± 0.04	0.7910 ± 0.04	0.7873 ± 0.04	0.7921 ± 0.04	0.7793 ± 0.04
Synthetic1	0.5284 ± 0.03	0.5607 ± 0.06	0.5651 ± 0.05	0.5622 ± 0.05	0.5697 ± 0.04	0.5675 ± 0.05
Synthetic2	0.7183 ± 0.02	0.7846 ± 0.02	0.7780 ± 0.02	0.7821 ± 0.02	0.7800 ± 0.02	0.7723 ± 0.02
UCI-Breast-Cancer	0.9643 ± 0.01	0.9722 ± 0.01	0.9735 ± 0.01	0.9734 ± 0.01	0.9719 ± 0.01	0.9727 ± 0.01

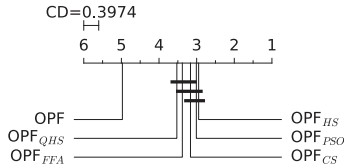


Fig. 4. Comparison of standard OPF against OPF ensemble pruning using metaheuristic techniques with the Nemenyi test in regarding the accuracy results. Groups of techniques that are not significantly different (at $p=0.05$) are connected.

this case, all pruning-based approaches belong in the same group (most accurate) according to the Nemenyi test.

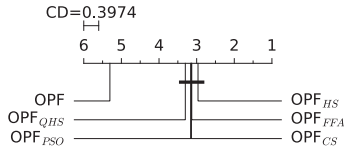
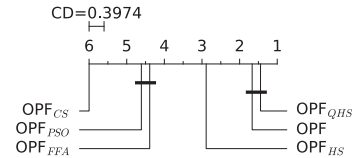


Fig. 5. Comparison of standard OPF against OPF ensemble pruning using metaheuristic techniques with the Nemenyi test in regarding the F-measure results. Groups of techniques that are not significantly different (at $p=0.05$) are connected.

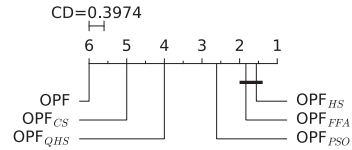
Table IV shows the mean computational load of all compared techniques with respect to the training step for ensemble pruning using 9 classifiers. The values in bold stand for the most accurate techniques according concerning the Wilcoxon signed-rank test. For all metaheuristic technique (i.e., CS, FFA, HS, PSO, QHS), the training time includes the training phase + validating step to finding the optimal sub-ensemble. The QHS has been considered the fastest one in almost of datasets. Figure 6a depicts the computational load (seconds) considering the training time (training+validating) with the Nemenyi test for ensemble pruning using 9 classifiers. The group considered the fastest one in the training+validating phase consists of standard OPF and OPF pruning using QHS, wherewith there is no CD between them. On average, the OPF pruning using QHS has been about 1.351 times faster than standard OPF in training+validating phase, since QHS approach has shown a rapid convergence against other metaheuristic techniques, and faster for training compared to the traditional approach (notice

a larger training set - $D_1 \cup D_v$ - in standard OPF requires more computational effort).

In regard to the testing phase, Table V presents the mean computational load in seconds, and Figure 6b shows the Nemenyi test concerning all compared techniques. As we can observe, the HS and FFA approaches have been the fastest ones, since both techniques pruned more classifiers with respect to the original ensemble, i.e., fewer classifiers for testing phase. It is worth noting that QHS approach has demonstrated interesting results in the statistical test (Figure 4) according to the recognition accuracies, but being among the fastest ones for training+validating. Also, QHS was around 1.525 times faster than standard OPF considering the testing phase.



(a)



(b)

Fig. 6. Nemenyi statistical test regarding the computational load for: (a) training (training+validating) and (b) testing phases. Groups that are not significantly different (at $p = 0.05$) are connected to each other.

VI. CONCLUSIONS

In this work, we introduced the idea of ensemble pruning considering the OPF classifier, as well as a quaternion-based representation. Roughly speaking, the idea is to build a reduced ensemble of OPF classifiers using metaheuristic-based optimization algorithms, such as HS, PSO, CS, FFA and QHS. Experiments over real and synthetic datasets showed the robustness of the proposed approach, which obtained the best results in almost all datasets. The proposed approach has

TABLE IV
COMPUTATIONAL LOAD (IN SECONDS) CONCERNING STANDARD OPF AGAINST OPF WITH ENSEMBLE PRUNING USING METAHEURISTIC TECHNIQUES WITH RESPECT TO THE TRAINING TIME (TRAINING + VALIDATING).

Dataset	OPF _(baseline)	OPF _{CS}	OPF _{FFA}	OPF _{HS}	OPF _{PSO}	OPF _{QHS}
Pima-Indians-Diabetes	0.0193 ± 0.00	1.9026 ± 0.65	0.1855 ± 0.07	0.0466 ± 0.01	0.1906 ± 0.07	0.0111 ± 0.00
Statlog-Australian	0.0178 ± 0.00	1.5261 ± 0.50	0.1434 ± 0.05	0.0391 ± 0.01	0.1514 ± 0.05	0.0111 ± 0.00
Statlog-Heart	0.0040 ± 0.00	0.2626 ± 0.08	0.0291 ± 0.01	0.0088 ± 0.00	0.0279 ± 0.01	0.0101 ± 0.00
Synthetic1	0.0053 ± 0.00	0.7897 ± 0.29	0.0760 ± 0.03	0.0212 ± 0.01	0.0823 ± 0.03	0.0102 ± 0.00
Synthetic2	0.0244 ± 0.00	2.4892 ± 0.79	0.2710 ± 0.10	0.0622 ± 0.02	0.2684 ± 0.09	0.0115 ± 0.00
UCI-Breast-Cancer	0.0170 ± 0.00	1.0168 ± 0.37	0.1032 ± 0.03	0.0289 ± 0.01	0.1039 ± 0.03	0.0110 ± 0.00

TABLE V
COMPUTATIONAL LOAD (IN SECONDS) CONCERNING STANDARD OPF AGAINST OPF WITH ENSEMBLE PRUNING USING METAHEURISTIC TECHNIQUES WITH RESPECT TO THE TESTING TIME.

Dataset	OPF _(baseline)	OPF _{CS}	OPF _{FFA}	OPF _{HS}	OPF _{PSO}	OPF _{QHS}
Pima-Indians-Diabetes	0.00589 ± 0.001	0.00470 ± 0.001	0.00270 ± 0.001	0.00275 ± 0.001	0.00312 ± 0.001	0.00392 ± 0.001
Statlog-Australian	0.00465 ± 0.001	0.00380 ± 0.001	0.00222 ± 0.001	0.00234 ± 0.001	0.00243 ± 0.001	0.00312 ± 0.001
Statlog-Heart	0.00116 ± 0.000	0.00064 ± 0.000	0.00045 ± 0.000	0.00038 ± 0.000	0.00044 ± 0.000	0.00053 ± 0.000
Synthetic1	0.00242 ± 0.000	0.00190 ± 0.000	0.00107 ± 0.001	0.00124 ± 0.000	0.00137 ± 0.000	0.00159 ± 0.000
Synthetic2	0.00782 ± 0.001	0.00625 ± 0.002	0.00452 ± 0.001	0.00424 ± 0.001	0.00450 ± 0.001	0.00530 ± 0.001
UCI-Breast-Cancer	0.00325 ± 0.001	0.00250 ± 0.001	0.00148 ± 0.001	0.00156 ± 0.000	0.00165 ± 0.000	0.00207 ± 0.001

demonstrated to improve standard OPF classification results, as well as QHS has shown to hold the best trade-off between effectiveness and efficiency.

ACKNOWLEDGMENT

The authors are grateful to FAPESP grants #2014/16250-9 and #2014/12236-1, CNPq grant #306166/2014-3, as well as CAPES.

REFERENCES

- [1] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 27, no. 5, pp. 553–568, 1997.
- [2] H. Toman, L. Kovacs, A. Jonas, L. Hajdu, and A. Hajdu, "Generalized weighted majority voting with an application to algorithms having spatial output," in *Proceedings of the 7th international conference on Hybrid Artificial Intelligent Systems - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 56–67.
- [3] Y. Bi, D. Bell, H. Wang, G. Guo, and J. Guan, "Combining multiple classifiers using dempster's rule for text categorization," *Applied Artificial Intelligence*, vol. 21, no. 3, pp. 211–239, 2007.
- [4] L. Jia, Y. Wang, and L. Fan, "Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm," *Integrated Computer-Aided Engineering*, vol. 21, no. 1, pp. 77–90, 2014.
- [5] M. M. Joly, T. Verstraete, and G. Paniagua, "Integrated multifidelity, multidisciplinary evolutionary design optimization of counterrotating compressors," *Integr. Comput.-Aided Eng.*, vol. 21, no. 3, pp. 249–261, 2014.
- [6] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.
- [7] B. Li, L. Qi, M. Q.-H. Meng, and Y. Fan, "Using ensemble classifier for small bowel ulcer detection in wireless capsule endoscopy images," in *IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 2326–2331.
- [8] F. Markatopoulou, G. Tsoumakas, and I. Vlahavas, "Instance-based ensemble pruning via multi-label classification," in *22nd IEEE International Conference on Tools with Artificial Intelligence*, vol. 1, 2010, pp. 401–408.
- [9] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE TPAMI*, vol. 31, no. 2, pp. 245–259, 2009.
- [10] S. Sheen, S. V. Aishwarya, R. Anitha, S. V. Raghavan, and S. M. Bhaskar, *Hybrid Artificial Intelligent Systems: 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part II*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, ch. Ensemble Pruning Using Harmony Search, pp. 13–24.
- [11] B. Krawczyk, "One-class classifier ensemble pruning and weighting with firefly algorithm," *Neurocomputing*, vol. 150, Part B, pp. 490 – 500, 2015.
- [12] M. Jodavi, M. Abadi, and E. Parhizkar, "Jsobfusedetector: A binary pso-based one-class classifier ensemble to detect obfuscated javascript code," in *International Symposium on Artificial Intelligence and Signal Processing*, 2015, pp. 322–327.
- [13] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.
- [14] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.
- [15] J. P. Papa, S. E. N. Fernandes, and A. X. Falcão, "Optimum-path forest based on k-connectivity: Theory and applications," *Pattern Recognition Letters*, 2016.
- [16] M. P. Ponti and J. P. Papa, "Improving accuracy and speed of optimum-path forest classifier using combination of disjoint training subsets," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, vol. 6713, 2011, pp. 237–248.
- [17] M. P. Ponti, J. P. Papa, and A. L. M. Levada, "A markov random field model for combining optimum-path forest classifiers using decision graphs and game strategy approach," in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. Lecture Notes in Computer Science, 2011, vol. 7042, pp. 581–590.
- [18] J. Kennedy and R. Eberhart, *Swarm Intelligence*. M. Kaufman, 2001.
- [19] Z. W. Geem, *Music-Inspired Harmony Search Algorithm: Theory and Applications*, 1st ed. Springer Publishing Company, Incorporated, 2009.
- [20] J. P. Papa, D. R. Pereira, A. Baldassin, and X.-S. Yang, "On the harmony search using quaternions," in *2016 European Signal Processing Conference*, 2016, (submitted).
- [21] X.-S. Yang and S. Deb, "Engineering optimisation by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 1, pp. 330–343, 2010.
- [22] X.-S. Yang, "Firefly algorithm, stochastic test functions and design

- optimisation,” *International Journal Bio-Inspired Computing*, vol. 2, no. 2, pp. 78–84, 2010.
- [23] C. Allène, J.-Y. Audibert, M. Couprie, and R. Keriven, “Some links between extremum spanning forests, watersheds and min-cuts,” *Image and Vision Computing*, vol. 28, no. 10, pp. 1460–1471, 2010.
- [24] D. Eberly, “Quaternion algebra and calculus,” Magic Software, Tech. Rep., 2002.
- [25] I. Fister, X.-S. Yang, J. Brest, and I. F. Jr., “Modified firefly algorithm using quaternion representation,” *Expert Systems with Applications*, vol. 40, no. 18, pp. 7220–7230, 2013.
- [26] L. Xu, A. Krzyzak, and C. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 3, pp. 418–435, 1992.
- [27] R. Larkins and M. Mayo, “Adaptive feature thresholding for off-line signature verification,” in *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*, 2008, pp. 1–6.
- [28] F. Wilcoxon, “Individual Comparisons by Ranking Methods,” *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, Dec. 1945.
- [29] P. Nemenyi, *Distribution-free Multiple Comparisons*. Princeton University, 1963.
- [30] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.