Feature Selection With Harmony Search

Ren Diao and Qiang Shen

***Abstract*—Many search strategies have been exploited for the task of feature selection (FS), in an effort to identify more compact and better quality subsets. Such work typically involves the use of greedy hill climbing (HC), or nature-inspired heuristics, in order to discover the optimal solution without going through exhaustive search. In this paper, a novel FS approach based on harmony search (HS) is presented. It is a general approach that can be used in conjunction with many subset evaluation techniques. The simplicity of HS is exploited to reduce the overall complexity of the search process. The proposed approach is able to escape from local solutions and identify multiple solutions owing to the stochastic nature of HS. Additional parameter control schemes are introduced to reduce the effort and impact of parameter configuration. These can be further combined with the iterative refinement strategy, tailored to enforce the discovery of quality subsets. The resulting approach is compared with those that rely on HC, genetic algorithms, and particle swarm optimization, accompanied by in-depth studies of the suggested improvements.**

***Index Terms*—Feature selection (FS), harmony search (HS), meta-heuristics, parameter control.**

I. Introduction

THE MAIN aim of feature selection (FS) is to discover a minimal feature subset from a problem domain while retaining a suitably high accuracy in representing the original data [9]. Practical problems that arise when analyzing data in real-world applications are often related to the number of features (so-called "curse of dimensionality" [1]), and the inability to identify and extract patterns or rules easily due to high interdependence among individual features, or the behavior of combined features. Human evaluation and subsequent pattern identification are limited when considering data sets which have very large numbers of features [46], [57]. Techniques such as text processing and classification [30] can benefit greatly from FS once the noisy, irrelevant, redundant, or misleading features are removed.

Given a data set with n features, the task of FS can be seen as a search for an "optimal" feature subset through the competing 2^n candidate subsets. Optimality of subsets is subjective, depending on the problem at hand, and a subset that is selected as optimal using one particular evaluation function may not be equivalent to that of a subset selected by another. Various evaluation techniques have been developed in the literature to judge the quality of the discovered feature subsets, such as methods based on rough sets [6], [36], [44], [48], [50], [58] and fuzzy-rough sets [3], [24], [25], [51], probabilistic consistency [8], [34], and correlation analysis [19], [20]. The aforementioned techniques are often referred to as filter-based approaches which are usually used as a preprocessing step and are independent of any learning algorithm that may be subsequently employed. Wrapper methods [22], [28] in contrast to filter approaches are often used in conjunction with a learning or data mining algorithm, where the learning algorithm forms part of the validation process. The generalized wrapper algorithm is similar to the filter approach apart from the fact that a learning algorithm is employed in place of an evaluation metric as used in the filter approach. Hybrid algorithms [60] exist which combine the benefits provided by both types of approach. When classification tests must be undertaken in parallel, the work of Min *et al.* [40] can be used to reduce the cost of such tests.

To locate the "optimal" feature subset, an exhaustive method may be used; however, it is often impractical for most data sets. Alternatively, hill-climbing (HC)-based approaches are exploited where features are added or removed one at a time until there is no further improvement to the current candidate solution. These methods have the advantage of fast convergence, but the use of evaluation metric on an individual feature basis may lead to a strong assumption that features are entirely independent of each other. Potentially correlated groups of features may not receive as much attention, where such interfeature dependences are usually very common in real-world data. As a result, HC approaches may lead to the discovery of suboptimal subsets [33], both in terms of the evaluation score and the subset size. Other algorithms use random search or heuristic strategies in an attempt to avoid such shortcomings; nature-inspired heuristics such as genetic algorithms (GAs) [29], [56], genetic programming [41], tabu search [21], simulated annealing [11], and particle swarm optimization (PSO) [53] are utilized to FS with varying degrees of success.

Harmony search (HS) [18], [32] is a recently developed meta-heuristic algorithm that mimics the improvisation process of music players. The HS algorithm has been very successful in a wide variety of engineering optimization problems [16], [31], [49], [52], [59] and machine learning tasks [10], [38], [39], [45]. It has demonstrated several advantages over traditional optimization techniques. HS imposes only limited mathematical requirements and is not sensitive to the initial value settings. Although it is a population-based approach, HS works by generating a new vector that encodes a candidate solution, after considering a selection of existing quality vectors. This forms a contrast with conventional evolutionary approaches such as

Manuscript received August 26, 2011; revised December 15, 2011; accepted March 1, 2012. Date of publication May 23, 2012; date of current version November 14, 2012. This paper was recommended by Editor S. Hu.

The authors are with the Department of Computer Science, Aberystwyth University, SY23 3DB Aberystwyth, U.K. (e-mail: rrd09@aber.ac.uk; qqs@aber.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2012.2193613

GAs that consider only two (parent) vectors in order to produce a new (child) vector. It increases the robustness and flexibility of the underlying search mechanism and, hence, helps to obtain better solutions. However, the nature of predefined constant parameters limits the exploitation of the original HS algorithm. The original technique has been improved by methods to adjust its pitch adjustment rate and bandwidth with regard to search iterations [37]. Furthermore, fret width is introduced to replace the static valued bandwidth [18], making the algorithm more adaptive to the variance in variable range and more suitable to solve real-valued problems. Work has also been carried out to analyze the evolution of the population variance over successive generations in HS, thereby drawing important conclusions regarding the explorative power of HS [7].

In this paper, a novel HS-based FS (HSFS) approach is proposed which incorporates several improvements over the initial ideas [13]. The original HS uses static parameters which may be difficult to determine without several test runs. Using the same parameter setting for both the initial exploration and the final fine-tuning may also limit the search performance. To resolve these drawbacks, HS is enhanced by introducing methods to tune parameters dynamically: An initial setup is used to encourage exploration; the parameter values then gradually change over iterations. At the end of the search, a different setup is prepared for fine-tuning of the final result. In contrast to the fixed-parameter version, the effort spent in determining good parameter settings is reduced significantly, while the overall search performance is also improved. Furthermore, the original algorithm focuses on single objective optimization, where the problem domain of FS is at least 2-D (minimal subset size and optimal subset evaluation). In order to adapt to the problem domain of FS, the search process has been modified by exploiting an iterative refinement strategy that recursively searches for smaller feature subsets while preserving the evaluation quality.

The remainder of this paper is structured as follows. Section II introduces the main concepts of HS. Section III suggests the parameter control scheme that further improves the overall search process. Section IV describes how FS problem can be modeled as an optimization task solvable by HS and details the approaches developed to tackle the problem. An illustrative example given in Section IV-B uses the popular FS technique based on fuzzy-rough sets [25]. The proposed iterative refinement technique is explained in Section V. This allows HS to locate smaller feature subsets when the algorithm converges in terms of evaluation score, which is particularly beneficial for high-dimensional data sets. Section VI-A presents the experimentation carried out on real-world problem cases [42] and presents comparative results gathered from the proposed approach against those that rely on HC, GAs, and PSO. Classification results are also shown in Section VI-B, where popular classifier learners are trained using those selected feature subsets. The classifiers used in the experiments include C4.5 [54], vaguely quantified fuzzy-rough nearest neighbor (VQNN) [23], naive Bayes' rule (NB) [26], and support vector machine (SVM) [27]. Section VI-C reveals the differences in results when different parameter control rules are used. The effects of the proposed enhancements are then examined in Section VI-D with comparison against the original HS. A demonstration of iterative refinement strategy for finding more compact fuzzy-rough reducts is also included in Section VI-E. Finally, Section VII concludes this paper and proposes further work in the area.

II. Principles of HS

HS mimics the improvisation process of musicians during which each musician plays a note for finding a best harmony all together. When applied to optimization problems, the musicians typically represent the decision variables of the cost function, and HS acts as a meta-heuristic algorithm which attempts to find a solution vector that optimizes this function. In such a search process, each decision variable (musician) generates a value (note) for finding a global optimum (best harmony). The HS algorithm, therefore, has a novel stochastic derivative (for discrete variable) based on musician's experience, rather than gradient (for continuous variable) in differential calculus.

A. Key Concepts

The key concepts of HS algorithm are musicians, notes, harmonies, and harmony memory. In most optimization problems solvable by HS, the musicians are the decision variables of the function being optimized. The notes played by the musicians are the values each decision variable can take. The harmony contains the notes played by all musicians, namely a solution vector containing one value per variable. Harmony memory contains harmonies played by the musicians, or it can be viewed as the storage place for solution vectors.

A more concrete representation of harmony memory is a 2-D matrix, where the rows contain harmonies (solution vectors) and the number of rows is predefined and bounded by the size of harmony memory. Each column is dedicated to one musician; it not only stores the good notes previously played by the musician but also provides the pool of playable notes for future improvisations. In this paper, such a column will be referred to as the note domain for the musician.

Original HS uses five parameters, including three core parameters such as the size of harmony memory ($\mathbf{HMS}$), the harmony memory considering rate ($\mathbf{HMCR}$), and the maximum number of iterations K, and two optional ones such as the pitch adjustment rate ($\mathbf{PAR}$) and the adjusting bandwidth (later developed into fret width) ($\mathbf{FW}$). The number of musicians N is defined by the problem itself and is equal to the number of variables in the optimization function. The detailed usage of these parameters, along with the iteration steps, is illustrated in the following sections.

B. Iteration Steps and Algorithm Illustration

HS can be divided into two core phases, initialization and iteration, as shown in Fig. 1. A simple example problem [18] is used for a better illustration

Minimize

$$(a-2)^2+(b-3)^4+(c-1)^2+3 \qquad (1)$$

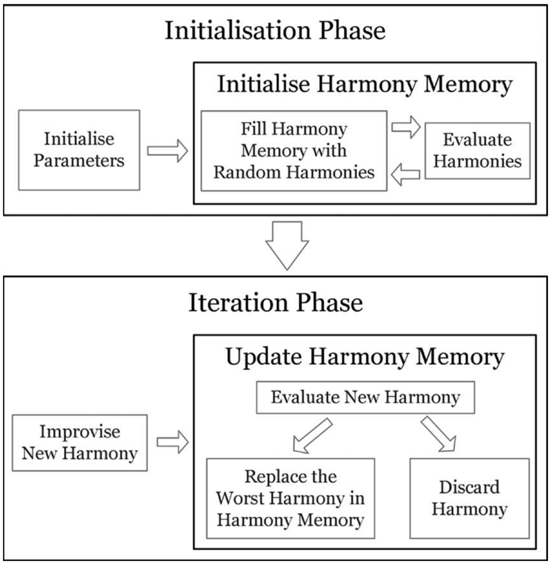


Fig. 1. Harmony search illustrated.

```
for ($j = 1$ to $N$)
    if ($random[0,1] <$ HMCR)
        $r := random\{1,$HMCR$\}$
        $Harmony_j := HM_{rj}$
        if ($random[0,1] <$ PAR)
            $\Delta_j = random[-1,1] \times$ FW$_j$
            $Harmony_j := Harmony_j + \Delta_j$
    else
        $Harmony_j := random[min_j, max_j]$
return $Harmony$
```

Fig. 2. HS improvisation process.

where

$$a, b, c \in \{1,2,3,4,5\}.$$

1) Initialize problem domain: In the beginning, the parameters used in the search need to be established. According to the problem at hand, the number of musicians is initialized to be equal to the number of variables (3), each corresponding to the decision attributes a, b, and c. Harmony memory is filled with randomly generated solution vectors. In the example problem, three randomly generated solution vectors may be $\{2,2,1\}$, $\{1,3,4\}$, and $\{5,3,3\}$.

2) Improvise new harmony: A new value is chosen randomly by each musician out of their note domain and together forms a new harmony. In the example, musician a may randomly choose 1 out of $\{2,1,5\}$, b chooses 2 out of $\{2,3,3\}$, and c chooses 3 out of $\{5,3,3\}$, forming a new harmony $\{1,2,3\}$.

   There are two factors which affect the note choice of a musician, namely, the harmony memory considering rate ($\mathbf{HMCR}$) and the pitch adjustment rate ($\mathbf{PAR}$). $\mathbf{HMCR}, 0 \leq \mathbf{HMCR} \leq 1$, is the rate of choosing one value from the historical notes stored in the harmony memory, while $(1 - \mathbf{HMCR})$ is the rate of randomly selecting one value from the range of all possible values. If $\mathbf{HMCR}$ is set low, the musicians will focus on exploring other areas of the solution space, and a high $\mathbf{HMCR}$ will restrict the musicians to historical choices.

   The $\mathbf{PAR}$, $0 \leq \mathbf{PAR} \leq 1$, parameter causes the musicians to select a neighboring value based on the formula $a + (random(-1,1) \times \mathbf{FW})$. For discrete variables, this simply means to choose the immediate left or right neighboring value. For continuous problems, $\mathbf{FW}$ is an arbitrary bandwidth that constrains the maximal amount of distance allowed to shift the current value. $(1 - \mathbf{PAR})$ is the probability of using the chosen value without further alteration. This pitch adjustment procedure only occurs if the note was chosen from the harmony memory (not affected by $\mathbf{HMCR}$).

   Given the aforementioned example, with $\mathbf{HMCR} = 0.9$ and $\mathbf{PAR} = 0.1$, musician a will choose from within the harmony memory $\{2,1,5\}$ with a probability of 90%. After making a choice, for example, 5 out of harmony memory, the musician will choose the left or right neighbors with 5% probability each, and the left neighboring value 4 may then be chosen in the end. Alternatively, the musician may choose from the range of all possible values $\{1,2,3,4,5\}$ with a probability of 10%, and the note 4 may again be chosen but without further pitch adjustment.

3) Update harmony memory: If the new harmony is better than the worst harmony in the harmony memory, judged by the objective function, the new harmony is then included in the resulting harmony memory, and the existing worst harmony is removed. The new harmony $\{1,2,3\}$ has the evaluation score of 9, making it better than the worst harmony in the memory $\{1,3,4\}$ which has a score of 16; therefore, the harmony $\{1,3,4\}$ is removed from memory, replaced with $\{1,2,3\}$. If $\{1,2,3\}$ had a larger score than 16, it would be the one being discarded.

   The algorithm continues to iterate until the maximum number of iterations K is reached. In the example, if the musicians later choose $\{2,3,1\}$, which is likely as those numbers are already in the note domains, the problem will be solved with a minimal fitness score of 3.

To further ease the understanding of HS, Fig. 2 shows an outline of the improvisation procedure in pseudocode. The aforementioned harmony memory is represented as 2-D matrix HM_{ij} of $\mathbf{HMS}$ rows and N columns. Here, $\mathbf{HMS}$ is the size of the harmony memory, and N is the number of musicians (the number of variables in the function). The note stored in the cell HM_{ij} belongs to the ith harmony and is played by the jth musician. $\min_j$ and $\max_j$ define the range of the jth variable.

C. Probabilistic View of HS

In order to demonstrate the convergence capability of HS, consider the harmony memory with the following parameters: the size of the harmony memory (the number of harmonies stored) $= \mathbf{HMS}$, the number of musicians (variables) $= N$, the number of possible notes (values) of a musician $= L$, the number of optimal notes (values) of musician j in the harmony memory $= H_j$, $\mathbf{HMCR}$, and the optimal harmony (optimal vector) $= (x, y, z)$. The probability of finding the optimal harmony $Pr(H)$ is

$$Pr(H) = \prod_{i=1}^{N} \left[\mathbf{HMCR} \frac{H_j}{\mathbf{HMS}} + (1 - \mathbf{HMCR}) \frac{1}{L} \right] \qquad (2)$$

where $\mathbf{PAR}$ is not considered because it is an optional operator.

Initially, the harmony memory is filled with random harmonies. If there is not any optimal note of all musicians in the harmony memory

$$H_1 = H_2 = \cdots = H_N = 0$$
$$Pr(H) = \left[(1 - \mathbf{HMCR}) \frac{1}{L} \right].$$

This means that the probability $Pr(H)$ is very low. However, if the schema of optimal harmony such as $(*, y, z)$, $(x, *, z)$, and $(x, y, *)$ have better evaluation than the others, the number of optimal notes of musician j in the harmony memory H_j will be increased iteration by iteration. Consequently, the probability of finding the optimal harmony $Pr(H)$ will be increased.

III. Parameter Control in HS

Traditional HS uses fixed predefined parameters throughout the entire search process, making it hard to determine a "good" setting without extensive trial runs. The parameters are also nonindependent from each other; therefore, finding a good setting often becomes an optimization problem itself. The search results usually provide no hint on how parameters should be adjusted in order to gain a performance increase. To eliminate the drawbacks lying with the use of fixed parameter values, a dynamic parameter adjustment scheme is proposed to modify parameter values at run time. By using tailored sets of parameter values for the initialization, intermediate, and termination stages, the search process can benefit greatly from this dynamic parameter environment.

At the beginning of a search, as the musicians are just starting to explore the solution space, the note domains contain only randomly initialized low-quality notes. Therefore, having a large harmony memory is not essential. In fact, having to keep a large pool of suboptimal harmonies may only confuse the musicians, preventing them from choosing good values during improvisation. Lower $\mathbf{HMCR}$ at this stage may also encourage the musicians to seek values outside of the current harmony memory.

As the search approaches the end, the musicians will usually have found many suboptimal harmonies. For such cases, given a high $\mathbf{HMCR}$, they will almost exclusively choose values out of the harmony memory when improvising new harmonies. Thus, a large pool of good results may contribute to a better solution. Of course, situations can also occur where the algorithm has not converged by the end of the search, which could be caused by the complexity of the problem itself or a less-than-desired number of iterations. From the earlier observation, a good dynamic $\mathbf{HMS}$ can be defined as

$$\mathbf{HMS}_K = \mathbf{HMS}_{\min} + \frac{K}{K_{\max}} (\mathbf{HMS}_{\max} - \mathbf{HMS}_{\min}).$$

TABLE I
Parameter Settings in Different Search Stages

	Initialisation	Intermediate	Termination
HMCR	Small	Medium	Large
HMS	Small	Medium	Large
Effect	High diversity. Deep Exploration	Steady improvement in harmonies	Fine tuning. Fast convergence

HS with a low $\mathbf{HMCR}$ takes less consideration of the historical values but focuses more on the entire value range when improvising new harmonies. HS with a high $\mathbf{HMCR}$ attempts to produce a new harmony out of existing values stored within the harmony memory. A dynamic $\mathbf{HMCR}$ that increases its value as the search progresses can be formulated such that

$$\mathbf{HMCR}_K = \mathbf{HMCR}_{\min} + \frac{K}{K_{\max}} (\mathbf{HMCR}_{\max} - \mathbf{HMCR}_{\min}).$$

Because one of the main advantages of HS is its simplistic structure, the specification of the rules is designed in this manner by taking the computational complexity into consideration. The calculus involved is made as simple as possible. Alternatively, smooth exponentially increasing functions may be considered

$$\mathbf{HMS}_K = \mathbf{HMS}_{\min} + 2^{\frac{K}{K_{\max}}-1} (\mathbf{HMS}_{\max} - \mathbf{HMS}_{\min})$$
$$\mathbf{HMCR}_K = \mathbf{HMCR}_{\min} + 2^{\frac{K}{K_{\max}}-1} (\mathbf{HMCR}_{\max} - \mathbf{HMCR}_{\min}).$$

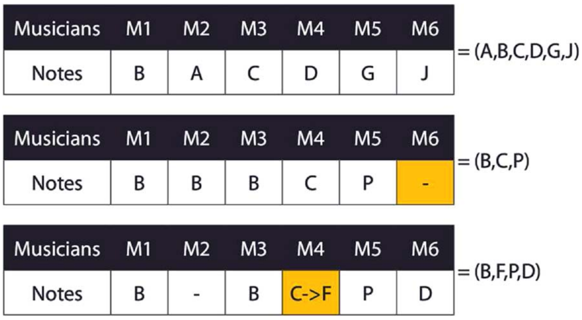
Although an exponentially decreasing function was proposed to control the fret width [37], for the general scenario of FS problem discussed in this paper, it is counterintuitive to suggest that, in any search stage, $\mathbf{HMS}$ and $\mathbf{HMCR}$ parameters should be adjusted in a more aggressive manner than the others.

All aforementioned individual parameter adjustment strategies can be combined together for a greater performance gain, allowing different sets of parameter settings for different search stages, as summarized in Table I. After initialization, the algorithm employs a large harmony memory, with a large chance of randomly selecting new values. Toward the intermediate stage, the algorithm uses a medium-sized harmony memory, with a balanced possibility between choosing values from the harmony memory and the range of all possible values. Finally, toward the termination of the process, the algorithm utilizes a small harmony memory, with the values chosen almost purely from stored good solutions. Note that these stages are listed here for conceptual explanation purposes; there are no clear boundaries in between them in implementation. Parameter settings gradually shift from one stage to another as the search progresses.

To further justify these intuitive rules, in Section VI-C, results are gathered and compared against the original algorithm with no parameter adjustments, as well as the algorithm using opposite rules, such that $\mathbf{HMS}$ and $\mathbf{HMCR}$ decrease from a maximum to minimum value over iterations.

TABLE II
CONCEPT MAPPING FROM HARMONY SEARCH TO FEATURE SELECTION

Harmony Search	Optimisation	Feature Selection
Musician	Variable	Feature Selector
Note	Variable Value	Feature
Harmony	Solution Vector	Subset
Harmony Memory	Solution Storage	Subset Storage
Harmony Evaluation	Fitness Function	Subset Evaluation
Optimal Harmony	Optimal Solution	Optimal Subset



Musicians	M1	M2	M3	M4	M5	M6
Notes	B	A	C	D	G	J

= (A,B,C,D,G,J)

Musicians	M1	M2	M3	M4	M5	M6
Notes	B	B	B	C	P	-

= (B,C,P)

Musicians	M1	M2	M3	M4	M5	M6
Notes	B	-	B	C->F	P	D

= (B,F,P,D)

Fig. 3. Harmony encoded feature subsets.

IV. HS FOR FS

In this section, a description of HSFS is given, based on the initial work [13]. It explains how FS problems can be translated into optimization problems, further solved by HS. This section includes illustrative examples of the encoding scheme used to convert feature subsets into harmony representation. A flow diagram of the search process is also shown in Fig. 4 along with step-by-step descriptions using fuzzy-rough FS (FRFS) as an example subset evaluator.

For the clarity of presentation, once the concept mapping has been introduced, the subsequent part of this paper will use the FS terms instead of HS concepts.

A. Key Concept Mapping

For conventional optimization problems, the number of variables is predetermined by the function to be optimized. However, for FS, there is no fixed number of features in a subset. The size of the emerging subset itself should be reduced in parallel to the optimization of the subset evaluation score. Therefore, when converting concepts, as shown in Table II, a musician is best described as an independent expert or "feature selector," where the available features for the feature selectors translate to notes for musicians. Each musician may vote for one feature to be included in the feature subset when such an emerging subset is being improvised. The harmony is then the combined vote from all musicians, indicating which features are being nominated. The entire pool of the original features forms the range of notes available to each of the musicians. Multiple musicians are allowed to choose the same attribute, and they may opt to choose no attribute at all. The fitness function used will become a feature subset evaluation method, which analyzes and merits each of the new subsets found during the search process.

For example, as shown in Fig. 3, the harmony $\{B, A, C, D, G, J\}$ represents a subset of size 6, where all musicians decided to choose distinctive notes. The second harmony $\{B, B, B, C, P, -\}$ demonstrates a duplication of choices from the first three musicians, and a discarded note (represented by -) from the last, resulting in a much reduced subset $\{B, C, P\}$ of size 3. The last harmony $\{B, -, B, C \rightarrow F, P, D\}$ will translate into feature subset $\{B, F, P, D\}$, where $C \rightarrow F$ indicates that the original vote from musician 4 was for C, but it was forced to change into F by **HMCR** activation.

For conventional optimization problems, the range of possible note choices for each musician is, in general, different from those for the other musicians. However, when applied to FS, all musicians jointly share one single value range, which is the set of all features.

B. HS Applied to FRFS

FRFS [25] is concerned with the reduction of information or decision systems through the use of fuzzy-rough sets. A fuzzy-rough set [15] is defined by two fuzzy sets, fuzzy lower and upper approximations, obtained by extending the corresponding crisp rough set notions [47]. In the crisp case [43], elements either belong to the lower approximation (i.e., have a membership of 1) or not. Given a data set with discretized attribute values, a subset (termed *reduct*) of the original attributes can be found, which are the most informative, by manipulating the rough sets defined over a data set; all other attributes can be removed from the data set with minimal information loss. In the fuzzy-rough case, elements may have a membership in the range [0, 1], allowing greater flexibility in handling uncertainty. Let $I = (\mathbb{U}, \mathbb{A})$ be an information system, where $\mathbb{U}$ is a nonempty set of finite objects (the universe) and $\mathbb{A}$ is a nonempty finite set of attributes such that $a : \mathbb{U} \rightarrow V_a$ for every $a \in \mathbb{A}$. V_a is the set of values that attribute a may take. For decision systems, $\mathbb{A} = \{\mathbb{C} \cup \mathbb{D}\}$, where $\mathbb{C}$ is the set of input features and $\mathbb{D}$ is the set of decision features. The following defines the fuzzy lower and upper approximations:

$$\mu_{\underline{R_P}X}(x) = \inf_{y \in \mathbb{U}} I\left(\mu_{R_P}(x, y), \mu_X(y)\right) \tag{3}$$

$$\mu_{\overline{R_P}X}(x) = \sup_{y \in \mathbb{U}} T\left(\mu_{R_P}(x, y), \mu_X(y)\right) \tag{4}$$

where I is a fuzzy implicator, T is a t-norm, and R_P is the fuzzy similarity relation induced by the subset of features P

$$\mu_{R_P}(x, y) = T_{a \in P}\left\{\mu_{R_a}(x, y)\right\} \tag{5}$$

with $\mu_{R_a}(x, y)$ being the degree to which objects x and y are similar for feature a. FRFS employs a quality measure termed the fuzzy-rough dependence function γ' to choose which features to add to the current reduct candidate, which is defined by

$$\gamma'_{P(Q)} = \frac{\sum\limits_{x \in \mathbb{U}} \mu_{POS_{R_P}(Q)}(x)}{\|\mathbb{U}\|} \tag{6}$$

where the fuzzy positive region is defined as

$$\mu_{POS_{R_P}(Q)}(x) = \sup_{q \in \mathbb{Q}} \mu_{\underline{R_P}Q}(x). \tag{7}$$

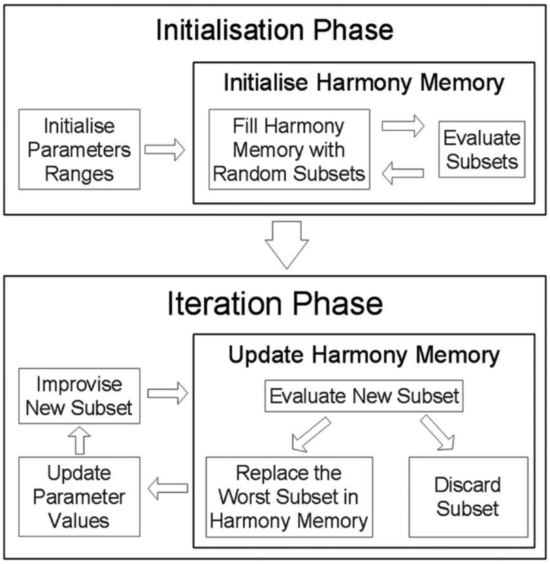


Fig. 4. Parameter-controlled harmony search applied to feature selection.

The original FRFS uses an HC-based algorithm termed fuzzy-rough QuickReduct, which is extended based on its crisp version [47]. Although the algorithm generally converges to a good solution fairly quickly, it only examines one feature at a time and disregards the potential correlations or interfeature dependences that may be present in the data. As a result, possible feature pairs or groups that jointly form an informative feature subset may be ignored simply because they do not individually contribute as much to the overall evaluation score. The algorithm terminates when the addition of any remaining feature does not increase the dependence.

C. Iteration Steps of HS-Based FRFS

The iteration steps of HS using the fuzzy-rough dependence function as the subset evaluator are demonstrated as follows, accompanied by the flow diagram shown in Fig. 4.

1) Initialize problem domain: The parameters are assigned according to the problem domain, including the following: $\mathbf{HMS}$, number of feature selectors N, maximum number of iterations K, and $\mathbf{HMCR}$. The subset storage containing $\mathbf{HMS}$ randomly generated subsets is then initialized. This provides each feature selector a working domain of $\mathbf{HMS}$ features, which may include identical choices, and nulls.
2) Improvise new subset: A new feature is chosen randomly by each feature selector out of their working feature domain and together forms a new feature subset. In the event of $\mathbf{HMCR}$, a random feature will be chosen from all available features to substitute the feature selector's own choice. For FS problems that are dealt with in this paper, the $\mathbf{PAR}$ parameter is not used. This is because the underlying motivation for the use of $\mathbf{PAR}$ is that minor adjustments into neighboring values may help discover better solutions, which is generally true for real-valued optimization functions. However, as the variables are now feature indices, each feature and its neighbors have no such general relations; thus, pitch adjustment will result in a change into an unrelated feature nearby.
3) Update subset storage: If the newly obtained subset achieves better fuzzy-rough dependence score than the worst subset in the subset storage, the new subset is included in the subset storage, and the existing worst subset is removed. The comparison of subsets takes into consideration of both dependence score and subset size in order to discover the minimal fuzzy-rough reduct at termination.

HS offers a clear advantage that a group of features is being evaluated as a whole. A newly improvised subset does not necessarily get included in the subset storage, just because one of the features has a locally strong fuzzy-rough dependence score. This is the key distinction to any HC-based approaches.

D. Complexity Comparison

For fuzzy-rough QuickReduct, given a data set of N features, the worst case complexity will result in $(N^2+N)/2$ evaluations of the dependence function. When implemented using HSFS, the number of subset evaluations is the same as the maximum number of iterations K, which is no longer dependent on the number of features in the original data. This characteristic makes HS more favorable when solving complex problems with large amount of features. As for the complexity of the HS algorithm itself, the initialization requires $O(N \times \mathbf{HMS})$ operations to randomly fill the subset storage, and the improvisation process is of the order $O(N \times K)$ because every feature selector needs to produce a new feature at every iteration. Here, $\mathbf{HMS}$ is the subset storage size, N is the number of feature selectors, and K is the maximum number of iterations. When comparing the storage requirement, HSFS clearly uses more storage as it needs to keep $O(N \times \mathbf{HMS})$ features in the subset storage, while HC only works on the current candidate solution, therefore requiring $O(N)$ storage space.

Although the two types of approach are analyzed here for the sake of comparison, in reality, FS is used for dimensionality reduction prior to any involvement of a given application which will exploit those features belonging to the resultant reduct. Thus, this operation has no negative impact upon the run-time efficiency of any subsequent process that utilizes the selected features.

V. Iterative Refinement for Finding More Compact Subsets

The parameter control technique previously introduced in Section III offers ways to dynamically change HS parameters in order to avoid some of the difficulties in finding a good set of parameters. However, there is an additional parameter introduced in the HSFS approach: the number of feature selectors N. For conventional optimization problems, N is equal to the number of variables in optimization function, which is predefined, and it restricts the number of columns in the subset storage. Due to the concept mapping in HSFS, the number of function variables is transformed into a virtual concept, and N

```
$\mathbb{S} \leftarrow$ ITERATIVEREFINEMENT($\mathbb{C}$,$\mathbb{D}$)
$\mathbb{C}$, the set of all conditional features;
$\mathbb{D}$, the set of decision features;
$\mathbb{S}$, the set of selected features.

$\Delta_{best} \leftarrow -\infty$, $\mathbb{S}_{best} \leftarrow \mathbb{C}$
$\mathbb{S} \leftarrow HarmonySearch(\mathbb{C}, |\mathbb{C}|)$
while ($\Delta(\mathbb{S}) \geq \Delta_{best}$)
    if (($\Delta(\mathbb{S}) == \Delta_{best}$)&&($|\mathbb{S}| == |\mathbb{S}_{best}|$))
        break
    else
        $\Delta_{best} \leftarrow \Delta(\mathbb{S})$
        $\mathbb{S}_{best} \leftarrow \mathbb{S}$
        $\mathbb{S} \leftarrow HarmonySearch(\mathbb{C}, |\mathbb{S}_{best}|)$
return $\mathbb{S}_{best}$
```

Fig. 5. Iterative refinement algorithm.

now serves as a hard upper bound for the resulting subset size, which needs to be defined by the user. Intuitively, N should be equal to the actual limit, the total number of features. Yet, such configuration often leads to less satisfactory results. This is because the current structure of HS only supports single objective optimization. Additional measures are required to enforce size reduction after HS converges in terms of the subset evaluation score. An alternative method would be to manually initialize N to a smaller value in order to force HSFS to find solutions within the restricted boundary. However, such an approach introduces human assumption prior to the search, and it is often difficult to estimate the amount of redundancy present in any given data set.

In order to combat this issue, an iterative refinement approach is proposed here such that the search process becomes more data driven, and to further reduce the need of manual parameter configuration. As shown in Fig. 5, the refinement process essentially performs HSFS iteratively, each time with a reduced feature selector size N. If a better or smaller subset is discovered in the previous iteration, the number of feature selectors is set to be equal to this subset's size. The refinement terminates when the current solution provides no improvement in either subset quality or size; the discovered solution is then returned.

VI. Experimentation and Discussion

In this section, the results of a number of experimentations carried out are reported to demonstrate the capabilities of the proposed approach: parameter-controlled HS with iterative refinement (HS-IR). A cross comparison against other search approaches is given in Section VI-A, including GAs [56], PSO [53], and HC, where both GAs and PSO are evolutionary global optimization heuristics. In order to demonstrate the generalization ability of HS-IR, three different subset evaluators are used in the experiments: probabilistic-consistency-based FS [8], correlation-based feature subset selection (CFS) [19], and FRFS [25] which was previously used as the example application in Section IV-B. In this research, the quality of the discovered subsets, or the performance of the search approaches, is judged by the subset evaluation score, in conjunction with the size of subset. This is, of course, independent of the actual classification accuracy measurement of the subsequent classifier that employs the selected feature subset for any filter-based approaches. However, for comparison, the classification accuracies tested using C4.5 [54], VQNN [23], NB [26], and SVM [27] are also shown in Section VI-B.

TABLE III
Data Set Properties

Dataset	Features	Instances	Decisions
ionosphere	35	230	2
water	39	390	3
waveform	41	5000	3
sonar	61	208	2
ozone	73	2534	2
libras	91	360	15
arrhythmia	280	452	16
secom	591	1567	2
isolet	618	7797	26
multifeat	650	2000	10

TABLE IV
Parameter Settings

Algorithm	Parameter	Values
Harmony Search Original (HS-O)	Memory Size	20
	Max Iteration	2000
	HMCR	0.8
	# Musicians	10
Harmony Search with Parameter Control (HS-PC)	Memory Size	10-20
	Max Iteration	1500
	HMCR	0.5-1
	# Musicians	10
Harmony Search with Parameter Control and Iterative Refinement (HS-IR)	Memory Size	10-20
	Max Iteration	1000
	HMCR	0.5-1
Genetic Algorithm (GA)	Cross Over Prob	0.6
	Max Generation	2000
	Mutation Prob	0.033
	Population Size	20
	Cross Over	0.6
Particle Swarm Optimisation (PSO)	C1	2.0
	C2	2.0
	Max Generation	1000
	# Particles	40

The experiments are performed on a selection of real-valued *UCI* benchmark data sets [42], a number of which are very large in size and high in dimension and hence present significant challenges to FS. A summary of the characteristics of these data sets is given in Table III. The parameter settings employed in the experiments are also summarized in Table IV. These parameter values are configured such that the population size and maximum number of iterations (or generations) are comparable with each other. In order to ensure convergence for the more complex data sets, a large number of iterations are uniformly chosen. Owing to the performance increase brought by parameter control and iterative refinement, the improved HS algorithm no longer requires as many iterations as the original to achieve good results. The maximum number of iterations used by HS-IR is therefore reduced by half of the original amount.

Stratified tenfold cross-validation (10-FCV) is employed for data validation. In 10-FCV, a given data set is partitioned into ten subsets. Of these ten subsets, nine subsets are used to perform a training fold, where FS algorithms are used to select the feature subsets. A single subset is retained as the

TABLE V
COMPARISON USING THE CONSISTENCY-BASED EVALUATOR ACROSS 10 × 10 CROSS-VALIDATION, REGARDING AVERAGED SUBSET SIZE, EVALUATION SCORE, AND TIME TAKEN. v, $-$, AND $*$ INDICATE STATISTICALLY BETTER, SAME, AND WORSE RESULTS

		HS-IR					GA			PSO			HC		
Dataset	Full	Size	t	Eval.	t	Time	Size	Eval.	Time	Size	Eval.	Time	Size	Eval.	Time
ionosphere	35	6.44	v	0.997	-	301.3	10.08	0.997	360.7	8.16	0.997	263.4	7.2	0.996	19.1
water	39	10	v	0.995	-	634.9	10.24	0.995	3315.9	12.68	0.995	561.7	10.9	0.995	64.9
waveform	41	10.62	v	1.000	-	45148	11.38	1.000	243348	12.54	0.999	27752	11.7	1.000	5396.3
sonar	61	11.46	v	0.990	v	552.4	12.06	0.990	4194.2	18.92	0.990	500.9	12.4	0.986	51.1
ozone	73	17.68	v	0.999	v	9274.3	19.62	0.999	78258.9	25.6	0.999	10068.1	20.1	0.929	5437.8
libras	91	15.54	-	0.972	v	1186.5	16.52	0.972	6118.3	26.58	0.972	1280.1	15.8	0.970	314.9
arrhythmia	280	22.38	-	0.988	-	3824.5	59.42	0.988	29155.2	111.38	0.988	6809.6	22.3	0.988	2677.9
secom	591	38.5	*	0.999	v	26923	180.85	0.999	180780	255.9	0.999	53021	1.1	0.933	867.3
isolet	618	12.84	-	1.000	-	128675	50.68	1.000	2283.4	16.83	1.000	89031	12.8	1.000	386542
multifeat	650	8.43	*	1.000	-	26033	49	1.000	546.3	13.76	1.000	18527	6.4	1.000	11727

TABLE VI
COMPARISON USING CFS EVALUATOR ACROSS 10 × 10 CROSS-VALIDATION, REGARDING AVERAGED SUBSET SIZE, EVALUATION SCORE, AND TIME TAKEN. v, $-$, AND $*$ INDICATE STATISTICALLY BETTER, SAME, AND WORSE RESULTS

		HS-IR					GA			PSO			HC		
Dataset	Full	Size	t	Eval.	t	Time	Size	Eval.	Time	Size	Eval.	Time	Size	Eval.	Time
ionosphere	35	10.06	-	0.542	-	54.3	10.38	0.542	126.2	11.06	0.537	113.7	10	0.542	5.1
water	39	10.1	-	0.427	-	35.6	10.38	0.427	87.2	8.68	0.419	87.2	10.2	0.427	3.1
waveform	41	14.9	-	0.384	-	344.2	14.9	0.384	436.2	12.41	0.369	508.6	14.9	0.384	207.9
sonar	61	17.22	*	0.359	-	60.6	18.02	0.359	186.2	12.02	0.325	127.3	18	0.359	7.3
ozone	73	19.9	v	0.114	-	386.2	21.58	0.114	604.5	25.34	0.110	463.3	21.3	0.114	160.8
libras	91	24.26	-	0.607	-	128.4	30.82	0.607	481.7	26.36	0.594	227.6	24	0.607	25.3
arrhythmia	280	27.36	v	0.441	v	1219.1	65.38	0.189	3379.4	13.48	0.279	956.7	24.3	0.437	197
secom	591	15.34	*	0.097	-	23719	96.3	0.007	36467	21.49	0.017	26289	13.3	0.100	2522.7
isolet	618	205.71	v	0.723	v	368224	275.26	0.721	453805	401.99	0.700	493006	183.3	0.710	271706
multifeat	650	124.11	*	0.910	*	57069	255.21	0.879	129125	358.09	0.849	63104	91.1	0.927	24876

testing data so that a classifier learner can be tested using the selected feature subsets. This cross-validation process is then repeated ten times (the number of folds). In the experiment, 10-FCV is performed ten times in order to lessen the impact of random factors within the heuristic algorithms; these 10 × 10 sets of evaluations are then aggregated to produce the final experimental outcomes. The advantage of 10-FCV over random subsampling is that all objects are used for both training and testing, and each object is used for testing only once per fold. The stratification of the data prior to its division into different folds ensures that each class label (as far as possible) has equal representation in all folds, thereby helping to alleviate bias/variance problems [2].

In addition to the comparative studies against the aforementioned approaches, further experiments are carried out in order to demonstrate the characteristics of HS. Section VI-C reveals the differences in results when different parameter control rules are used. This empirically demonstrates that the recommended set of rules as presented in Section III is both intuitively sound and practically effective. A comparison against the original HS algorithm is provided in Section VI-D in order to show the effects of the proposed enhancements: parameter control and iterative refinement. An illustration of how iterative refinement can further aid FRFS in discovering more compact fuzzy-rough reducts is also included in Section VI-E.

A. Comparison Against Other Search Methods

Paired t-test has been employed to evaluate the significance of differences between the algorithms' results. In Tables V–VII, paired t-test using $p = 0.01$ has been performed to evaluate the statistical differences between the results obtained by HS-IR and HC. The symbol v indicates that HS-IR obtained a better result than HC, $-$ denotes that there existed no differences between the results, and $*$ signifies that HS-IR resulted in statistically worse search performance. These comparisons are in terms of higher evaluation score or smaller subset size.

TABLE VII
COMPARISON USING THE FRFS EVALUATOR, REGARDING AVERAGED SUBSET SIZE AND EVALUATION SCORE BY 10 × 10 CROSS-VALIDATION. v, $-$, AND $*$ INDICATE STATISTICALLY BETTER, SAME, AND WORSE RESULTS

Dataset	Full	HC-IR		GA		PSO		HC	
ionosphere	35	6.1 v	1	9.94	1	7.3	1	7	1
water	39	6 -	1	8.34	1	6.68	0.998	6	1
sonar	61	5.48 *	1	8.66	1	6	0.997	5	1
libras	91	6.98 v	1	16.26	1	8.24	0.999	7.6	1
arrhythmia	280	7.17 -	1	28.28	1	9.62	1	7.1	1

Comparison Between Search Methods Using the Consistency-Based Evaluator: The consistency-based subset evaluator is used to produce the results shown in Table V. When comparing the results purely by evaluation scores, all three global optimization techniques (except PSO on waveform) discovered subsets with tied scores. However, HC is stuck at a local solution [33] in four out of ten occasions: sonar, ozone, libras, and secom, not reaching the globally obtained better scores. Although the absolute values in the differences seem to be minor, paired t-test results confirm that, statistically, HC consistently finds lower quality subsets across the 10 × 10 cross-validation process.

In terms of solution size, the proposed HS-IR approach finds more compact solutions in five out of ten instances, while HC finds smaller solutions for secom and multifeat. The results show that GAs and PSO can optimize the evaluation score but fail to reduce the subset size further, while HS-IR displays a clear advantage in terms of size reduction. The time taken for the search algorithms is also included in the table. GA generally uses the longest time, while HC consumes the least. For the larger data sets such as isolet and multifeat, while GA does not obtain the best solutions, its execution time is much lower when compared against the other approaches.

The results also indicate that HC approaches may be of a better performance in situations where the local solution happens to be the global one. However, it suffers from the local minima problem.

For example, for data set secom, HC selected the feature subsets with indices $\{38\}$ and $\{38, 153\}$ out of the 591 available features with an averaged evaluation score of 0.933 and then failed to progress further from this local solution. The other methods all succeeded in finding subsets scoring 0.999, but in various sizes.

Comparison Between Search Methods Using the CFS Evaluator: The results shown in Table VI are gathered using the CFS subset evaluator. For the first six data sets that have a lower dimensionality, all search techniques reached the same evaluation scores except PSO, where HS-IR discovered the smallest subsets for ozone and smaller subsets with better evaluation for sonar. When comparing the execution times, HS-IR shows an improvement over the other evolutionary algorithms.

Among the results obtained, HC located the best solutions for secom and multifeat but again encountered local minima problems for arrhythmia (0.437 versus 0.441) and isolet (0.710 versus 0.723). PSO generally found smaller subsets, but its evaluation scores are systematically lower than those achieved by the rest.

GA's performance is less stable for high-dimensional data sets, producing the second highest evaluation score for isolet but the lowest for arrhythmia and secom. Although HS-IR did not find the best solution for multifeat that was obtained by HC, the achieved solution is much superior than those by using the other two global optimization techniques.

Comparison Between Search Methods Using the FRFS Evaluator: FRFS is very effective at reducing redundant information while preserving the underlying semantics of data. However, it requires fuzzy similarity calculation between every pair of instances for every feature in the subset; as a result, the search process's complexity is to the order of $O(K \times M \times N^2)$, where K is the number of evaluations performed during the search, M is the size of the feature subset, and N is the number of instances. As such, for data sets with large number of instances such as secom, the number of required similarity computations within one search may exceed 4×10^{13}. This overhead on computational resources makes it difficult to perform 10 × 10-fold cross-validation over such large data sets in order to enable systematic evaluation as with the use of other FS methods. Thus, FRFS is herein only tested against a subset of data sets.

Results from the lower dimensional data sets, as shown in Table VII, indicate that the proposed approach can produce smaller optimal fuzzy-rough reducts. The optimal solution is obtained here in terms of the experimental evaluation because the full dependence score is reached. HS-IR discovered the most compact reducts in two out of five data sets: ionosphere and libras; HC found the smallest reduct for sonar. PSO again showed decent size reduction for most data sets; however, the evaluation scores are not always optimal. GA, on the other hand, found subsets with optimal evaluation but larger in size. While HC may work, the local solution problem discussed previously is again revealed in data sets ionosphere and libras.

TABLE VIII
COMPARISON OF MULTIPLE HS-IR REDUCTS VERSUS SINGLE HC REDUCT FOR THE ARRHYTHMIA DATA SET. ALL SUBSETS ARE OF SIZE 7 AND EVALUATION SCORE 1

	Feature Index
HSFS	3 5 9 80 242 246 275
	3 14 104 113 169 181 271
	14 20 64 209 238 242 251
	5 6 7 113 188 231 267
	3 6 76 161 246 252 265
	0 5 6 10 209 257 262
	14 191 208 218 241 259 275
	40 161 186 209 225 228 255
	6 80 128 161 208 239 271
	8 14 208 216 225 241 246
HC	0 3 6 168 169 217 251

The stochastic nature of global optimization techniques such as HS allows the discovery of multiple different solutions for the same set of training samples. Table VIII details the differences in the discovered reducts between HS-IR and HC using the data set arrhythmia as an example. These subsets are recorded during the experiment, where the same cross-validation fold is used by both methods. Note that similar properties have also been observed for other data sets, although the employment of arrhythmia allows such properties to be revealed more clearly. In general, different runs may converge to the same solution, possibly due to a limited number of best solutions that can be inferred from data sets of a lower dimensionality.

For ten runs of HS-IR, ten different reducts of an average size 7 are selected (again, all reaching the full dependence measure of 1), while HC results in a single subset of size 7.

The ability to produce multiple quality subsets from the same training data may greatly benefit multiview learning techniques such as classifier ensemble [14], [55], where the subsets may be used to generate partitions of the training data in order to build diverse classification models.

B. *Classification Accuracies*

The classifier learners C4.5 [54], VQNN [23], NB [26], and SVM [27] are employed for the purpose of illustrating the quality of subsets discovered in the FS phase. The same cross-validation training folds that were used for FS are used to evaluate the ten subsets from each search. This yields 10 × 10 results for each approach and each data set. The averaged classification accuracies testing on the corresponding testing folds, along with paired t-test results ($p = 0.01$), are shown in Tables IX–XI.

These results show that, after FS, the classification accuracy may vary when compared against the use of full features. Such results conform to the literature in that retaining more attributes in a data set may result in better approximations,

TABLE IX
C4.5 Classification Accuracies Using Feature Subsets Selected by the Consistency-Based Evaluator Versus That of Full Feature Subset, Regarding Averaged Subset Size and Evaluation Score by 10 × 10 Cross-Validation. v, $-$, and $*$ Indicate Statistically Better, Same, and Worse Results

Dataset	Full	HS-IR		GA		PSO		HC	
ionosphere	85.65	87.04	-	86.52	-	84.61	-	85.22	-
water	79.74	83.38	v	82.77	v	82.26	v	82.25	v
waveform	76.62	74.30	*	74.75	*	75.29	*	76.88	-
sonar	72.62	73.80	-	71.79	-	70.99	-	70.5	*
ozone	92.62	93.05	v	93.22	v	93.17	v	93.21	v
libras	70.28	65.33	*	65.22	*	67.39	*	65.83	*
arrhythmia	65.06	66.12	-	66.66	-	65.99	-	66.38	v
secom	88.96	92.58	v	90.62	v	90.51	v	90.24	v
isolet	83.42	50.37	*	66.94	*	51.59	*	60.73	*
multifeat	94.30	85.48	*	85.98	*	78.70	*	85.25	*

TABLE X
C4.5 Classification Accuracies Using Feature Subsets Selected by the CFS Evaluator Versus That of Full Feature Subset, Regarding Averaged Subset Size and Evaluation Score by 10 × 10 Cross-Validation. v, $-$, and $*$ Indicate Statistically Better, Same, and Worse Results

Dataset	Full	HS-IR		GA		PSO		HC	
ionosphere	85.65	85.30	-	85.22	-	85.57	-	85.21	-
water	79.74	82.46	v	82.36	v	81.18	v	82.56	v
waveform	76.62	77.23	v	77.22	v	77.19	v	77.22	v
sonar	72.62	72.95	-	72.48	-	72.74	-	73.14	-
ozone	92.62	93.28	v	93.31	v	93.47	v	93.49	v
libras	70.28	69.33	-	71.33	-	67.5	*	70.83	-
arrhythmia	65.06	67.27	v	66.74	v	63.19	-	66.81	-
secom	88.96	92.78	v	91.35	v	92.61	v	92.98	v
isolet	83.42	83.02	-	83.53	-	83.56	-	83.22	-
multifeat	94.30	94.63	-	94.34	-	94.37	-	94.15	-

TABLE XI
C4.5 Classification Accuracies using Feature Subsets Selected by the FRFS Evaluator Versus That of Full Feature Subset, Regarding Averaged Subset Size and Evaluation Score by 10 × 10 Cross-Validation. v, $-$, and $*$ Indicate Statistically Better, Same, and Worse Results

Dataset	Full	HS-IR	GA	PSO	HC
ionosphere	85.65	86.96 v	86.00 v	87.04 v	86.52 v
water	79.74	79.03 *	80.21 -	78.15 *	80.51 v
sonar	72.62	70.54 *	70.10 *	70.19 *	76.93 v
libras	70.28	60.39 *	64.06 *	56.67 *	61.11 *
arrhythmia	65.06	62.27 *	64.33 -	63.44 *	66.00 v

if the data set tends to be consistent with little noise and/or most of the features are independent. Feature dimensionality reduction is theoretically difficult in such situations without losing information. It may even be necessary to consider as many features as possible for certain application problems. However, the removal of features that are noisy, redundant, or irrelevant can improve the quality of the learning process. With such damaging features removed, classification performance can be improved. Additional benefits of using such reduced set of features include minimizing the measurement and storage requirements and reducing training and run time of the classifiers.

Comparison of Selected Subsets Using C4.5: Table IX compares the accuracy rates when C4.5 is used to test the reduced feature subsets discovered by the different search approaches, using consistency-based evaluator. For the water, ozone, and secom data sets, FS helped improve the overall classification accuracy, where, in four out of ten cases, the performance is decreased. Yet, it can also be observed that a higher subset evaluation score does not necessarily result in a more accurate learned classifier model. This is evident in the groups of subsets found by HS-IR and PSO which both scored 1, the same as the other two search methods, but the classification difference between them is as large as 10%.

Tables X and XI present results obtained using the feature subsets selected by CFS and FR. A direct comparison between the three subset evaluators' performances in terms of classification accuracy reveals that the CFS evaluator works very well at maintaining the amount of information in the reduced feature subset, although with larger sizes. FR can produce much smaller subsets but compromise classification performance slightly. Interestingly, the consistency-based evaluator is balanced between classification performance and size. The underlying reason for this remains an active investigation.

Comparison of Subsets Selected by HS-IR Using Different Classifiers: The results obtained by VQNN, NB, and SVM using the subsets obtained by HS-IR are condensed into Table XII to avoid duplication and save space. This is given in comparison to the use of the full set of features. Overall, the CFS performs the best in terms of preserving and improving classification accuracy. Out of the 30 different results obtained from the feature subsets selected by CFS, 13 cases show an increase in classification performance. For 8 of the remaining 17 cases, no statistical differences were observed between the use of full and reduced feature subsets. On the other hand, results obtained by consistency-based and FRFS reflect the loss of accuracy in most cases.

Note that, for data set ozone, the classifiers VQNN and SVM obtained identical cross-validated accuracy of 93.69%. Furthermore, the classifiers built using the reduced subsets discovered by both consistency-based and CFS also agree in terms of classification accuracy. A closer investigation into the selected feature subsets indicates that, among the 10 × 10 runs for ozone, the consistency-based method selected 47 different feature subsets, while CFS selected 49, with no common feature subsets between them. The underlying reason behind such similar performances remains an active research. The entire set of investigative experiments performed, over the impact of using reduced feature subset upon different classifiers, reflects the common understanding that FS and feature-based classification are independent issues as far as filter-based approaches are concerned [8], [28].

C. Comparison Between Parameter Control Rules

The parameter control rules discussed in Section III are examined here, with results compared against other possible approaches. The previously employed data sets are once again adopted, with either CFS or FRFS acting as the subset evaluator. For the less complex data sets, all algorithms lead to identical results, and therefore, those results are omitted. The following discussions focus on the remainder of the results. Here, the arrows illustrate how the parameters are adjusted over iterations, e.g., HMS $\searrow$ means that $\mathbf{HMS}$ decreases from maximum to minimum value as the search progresses; HMCR $\rightarrow$ means that $\mathbf{HMCR}$ is static throughout; and $\mathbf{HMS} \nearrow \mathbf{HMCR} \nearrow$ indicates the cases where both parameters increase over time, and hence the recommended rules.

TABLE XII
CLASSIFICATION ACCURACIES USING FULL FEATURE SUBSET VERSUS REDUCED FEATURES, REGARDING EVALUATION SCORES BY 10 × 10 CROSS-VALIDATION. v, $-$, AND $*$ INDICATE STATISTICALLY BETTER, SAME, AND WORSE RESULTS

Dataset	Naive Bayes				VQNN				Support Vector Machine			
	Full	CON	CFS	FR	Full	CON	CFS	FR	Full	CON	CFS	FR
ionoshpere	83.91	81.83 *	84.48 -	85.04 v	83.04	77.57 *	81.39 *	79.04 *	83.04	74.09 *	84.7 v	76.17 *
water	85.9	86.77 v	86.1 -	81.64 *	80.00	81.49 v	83.13 v	75.03 *	86.92	84.21 *	84.82 *	80*
waveform	80	79.46 -	67.38 *		68.54	70.17 v	77.31 v		86.7	81.74 *	87.33 v	
sonar	67.79	66.81 *	73.74 v	67.59 -	85.60	76.05 *	79.89 *	77.66 *	79.79	78.37 *	86.88 v	79.38 -
ozone	67.64	76.08 v	64.89 *		93.69	93.69 -	93.69 -		93.69	93.69 -	93.69 -	
libras	67.78	61.33 *	67.95 -	68.17 -	83.61	79.72 *	81.56 *	73.94 *	72.78	63.89 *	68.11 *	65.44 *
arrhythmia	61.07	66.56 v	80.52 v	68.2 v	55.98	53.04 *	55.78 -	52.71 *	70.37	66.06 *	67.71 *	66.14*
secom	29.75	58.34 v	77.04 v		93.36	92.34 *	91.67 *		92.92	63.36 *	93.36 v	
isolet	84.99	60.78 *	88.83 v		83.92	45.02 *	86.27 v		96.61	63.7 *	95.73 -	
multifeat	95.1	88.46 *	96.84 v		97.95	85.28 *	98.39 v		98.5	89.5 *	98.41 -	

TABLE XIII
COMPARISON OF PARAMETER CONTROL RULES USING CFS AVERAGED SUBSET SIZE ROUNDED TO NEAREST INTEGER AND EVALUATION SCORE BY 10 × 10 CROSS-VALIDATION

Mode	ionosphere		olitos		sonar		water 2		water 3	
	Size	Score	Size	Score	Size	Score	Size	Score	Size	Score
HMS↘ HMCR→	17	0.5020	15	0.5667	29	0.0907	16	0.1648	17	0.2545
HMS↗ HMCR→	15	0.5101	16	0.5676	28	0.1111	15	0.2638	16	0.3690
HMS→ HMCR→	16	0.5107	15	0.5675	27	0.1083	15	0.2441	16	0.3258
HMS→ HMCR↗	12	0.5155	16	0.5676	21	0.2978	11	0.3405	13	0.3940
HMS→ HMCR↘	15	0.5087	16	0.5676	26	0.1300	14	0.1944	16	0.2758
HMS↗ HMCR↗	12	0.5173	16	0.5677	20	0.2989	12	0.3409	13	0.4079
HMS↗ HMCR↘	15	0.5103	15	0.5665	27	0.1227	15	0.1819	15	0.2848
HMS↘ HMCR↗	17	0.5032	15	0.5673	30	0.0884	16	0.1735	17	0.2668
HMS↘ HMCR↘	16	0.5075	16	0.5676	28	0.1020	16	0.1887	16	0.2841

TABLE XIV
COMPARISON OF PARAMETER CONTROL RULES USING FRFS AVERAGED SUBSET SIZE ROUNDED TO NEAREST INTEGER ACROSS 10 × 10 CROSS-VALIDATION

Mode	ionosphere	olitos	sonar	water2	water3
HMS↘ HMCR→	14	9	29	16	16
HMS↗ HMCR→	12	7	25	14	14
HMS→ HMCR→	13	7	26	14	14
HMS→ HMCR↗	10	6	20	9	10
HMS→ HMCR↘	11	7	22	12	12
HMS ↗ HMCR↗	10	6	18	9	9
HMS↗ HMCR↘	12	7	22	13	13
HMS↘ HMCR↗	14	10	28	17	16
HMS↘ HMCR↘	13	8	27	14	14

Table XIII shows the subset size and evaluation score obtained using the CFS evaluator. The first three rows show different $\mathbf{HMS}$ adjustment functions with a static $\mathbf{HMCR}$. The effect of an increasing $\mathbf{HMS}$ can be spotted by the overall superior evaluation scores; the subset sizes are not differentiated by a great amount, but the decreasing $\mathbf{HMS}$ generally finds larger subsets. The third to fifth rows show the comparison of different $\mathbf{HMCR}$ functions with a static $\mathbf{HMS}$. Here, the main difference in results is the subset size, where an increasing $\mathbf{HMCR}$ helps to discover smaller subsets, and the evaluation score is also generally higher. Comparison of combined rule sets shows that the final results are better when both parameters are increasing during the search.

The same conclusion can be reached by studying the results obtained using FRFS as the subset evaluator, as shown in Table XIV. All HS variations have achieved full fuzzy-rough dependence measure for the discovered subsets; the difference in performance is therefore reflected purely by the size reduction. HS with increasing $\mathbf{HMS}$ and $\mathbf{HMCR}$ finds more compact subsets overall, while HS with a static $\mathbf{HMS}$ and increasing $\mathbf{HMCR}$ achieves a close second place with minor increase in subset size, once again demonstrating that $\mathbf{HMCR}$ adjustment plays a key role in reducing the subset size.

D. Effect of Parameter Control and Iterative Refinement

The effects of proposed improvements are demonstrated here with a comparison against the original HS algorithm. Table XV details the results obtained, showing both the subset size and the evaluation score. The columns labeled with HS-O contain subsets discovered by the original algorithm, and those labeled with HS-PC show the results of using parameter-controlled HS. HS-IR, which iteratively refines HS, is also included for comparison. For the purpose of maintaining consistency of evaluation, the selection process employs the same cross-validation folds as used in the previous sections. Paired t-tests are again employed to compare the differences between HS-PC and HS-O, and HS-IR against HS-PC. In all cases bar one, the enhancements offer statistically significant improvements in terms of size reduction and evaluation optimization. For data set secom, while HS-PC did not increase the evaluation score when compared to HS-O, it reduced the average subset size.

It can be seen from these results that the proposed improvements show greater effect under more complex situations, such as those involving subsets with larger amount of features and larger number of instances or containing many competing potential solutions. The effect of parameter control is mostly revealed in terms of better evaluation scores, while the subset sizes are also reduced in the process. However, iterative refinement greatly improves the overall solution quality and shows exceptional capability of reducing the size of subsets. For

TABLE XV
COMPARISON OF PROPOSED HS IMPROVEMENTS USING SUBSETS SELECTED BY CFS, REGARDING AVERAGED SUBSET SIZE, EVALUATION SCORE, AND C4.5 CLASSIFICATION ACCURACY, BY 10 × 10 CROSS-VALIDATION. v, $-$, AND $*$ INDICATE STATISTICALLY BETTER, SAME, AND WORSE RESULTS

Dataset	Full		HS-O			HS-PC			HS-IR		
ionoshpere	35	85.62	14.04	0.533	85.30	11.46	0.539	85.57 v	10.06	0.542	85.30 *
water	39	79.74	15.24	0.386	83.13	12.3	0.419	82.82 *	10.1	0.427	82.46 -
waveform	41	76.62	17.32	0.362	77.02	15.46	0.382	77.21 v	14.9	0.384	77.23 -
sonar	61	72.62	25.34	0.132	72.62	20.54	0.317	73.52 v	17.22	0.359	72.95 *
ozone	73	92.62	38.58	0.106	93.35	29.36	0.113	93.41 -	19.9	0.114	93.28 -
libras	91	70.28	51.12	0.582	70.56	43.4	0.603	70.83 v	24.26	0.607	69.33 *
arrhythmia	280	65.06	161.82	0.052	67.84	117.74	0.088	67.49 -	27.36	0.441	67.27 -
secom	591	88.96	348.78	0.002	90.06	279.64	0.002	90.74 v	15.34	0.087	92.78 v
isolet	618	83.42	383	0.692	83.37	356.98	0.716	83.59 -	205.71	0.723	83.02 *
multifeat	650	94.30	401.16	0.836	94.22	365.98	0.867	94.42 v	124.11	0.91	94.63 v

the secom data set, HS-IR succeeded in reducing the solution size by over 95% without sacrificing the evaluation score, making HS-IR a competitive algorithm in dealing with higher dimensional FS problems.

From the differences in classification accuracy, it can be seen that, out of ten data sets, HS-PC bears improvements over HS-O on six cases, ties for three cases, and underperforms on just one case. This demonstrates the effectiveness of using parameter control. HS-IR performs better when judged by the CFS evaluation scores and subset sizes. However, classification accuracy results indicate that more compact feature subsets (with equal or better evaluation score) may not necessarily lead to equal or better classifiers. For example, regarding data set arrhythmia, although HS-IR raised average evaluation score from 0.002 to 0.087 and reduced average subset size from 279.64 to 15.34, the classifier accuracy remained the same. Yet, the experimental results also show that, for this data set, HS-IR equipped with the CFS evaluator removed a fair amount of redundancy, while not affecting the end classifier performance.

E. Iterative Refinement for More Compact Fuzzy-Rough Reduct

Section VI-A3 showed that the iterative refinement technique works very well for finding smaller fuzzy-rough reducts. The following experimental results show graphically how an initial solution is improved over several refinements. Two data sets with a relatively large number of features are used: arrhythmia (280 features) and web (2557 features). The search objective is to find a fuzzy-rough reduct (with fuzzy-rough dependence measure of 1.0) of the minimal size. For the web data set, only ten different runs are performed due to its substantially high dimensionality; for each data set, the reduct sizes at each iteration are recorded, averaged, and summarized in Figs. 6 and 7, respectively.

For arrhythmia, the refinement is completed within five iterations in six out of ten runs, with an averaged final reduct size of 7.17. For web, 40% of the runs terminate within 30 refinements with the rest taking more than 33 iterations. For smaller data sets, more compact reducts are usually found within three iterations. In reality, if the search is to be performed multiple times, the efficiency can be further increased by limiting the starting feature selector size N to a smaller value, which may be discovered in the first few runs.

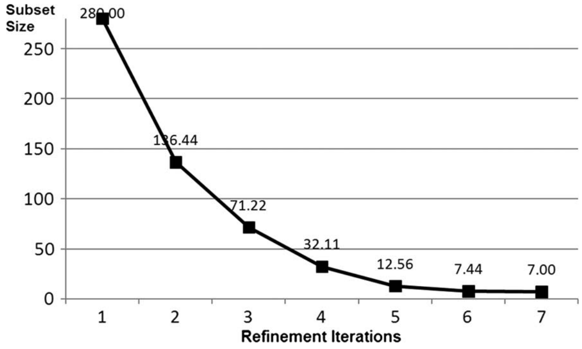


Fig. 6. Iterative fuzz-rough reduct refinement for arrhythmia data set.

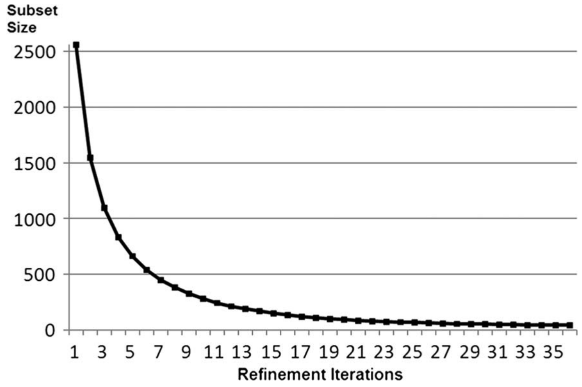


Fig. 7. Iterative fuzz-rough reduct refinement for web data set.

F. Discussion

HS-based approaches are computationally inexpensive themselves in terms of computation performance and robustness. This is because the algorithm comprises a very simple concept, and the implementation is also straightforward. The run time of the entire FS process is mainly determined by the following two factors: the max number of iterations K and the efficiency of the subset evaluation method. K can be manually configured according to the complexity of data set; in the experimental evaluation, HS converges very quickly with similar run time to that of GA and PSO. The experiments also revealed the downside of FRFS for being less scalable for larger data sets; empirically (and as may be expected), the larger the number of instances in the data set, the longer time is required for computing fuzzy-rough dependence measures.

The use of subset storage in HS offers a major advantage over that of techniques like GAs, as it maintains a record of the historical data processed by previously learned iterations. All elements of the memory together contribute to the new subset, while changes in genetic populations tend to result in the destruction of previous knowledge of the problem. $\mathbf{HMCR}$ also helps greatly for the search mechanism to escape from the local best solution. In all experiments, there has been no attempt to optimize the parameters for each of the search methods for comparison. The same parameter settings are used for easy comparison throughout regardless of the difference in complexities of the data sets. It can be expected that the results obtainable from the proposed work with optimization would be even better than those already observed.

The proposed approach offers an improved search heuristic. Unfortunately, in general, there is no search heuristic that can guarantee exhaustive search; otherwise, it is not a heuristic in the first place. Therefore, no subsets found can be theoretically proven to be a global optimal. However, in practice, it is important to investigate the relative strength of a given search heuristic. The systematical experimental studies presented in this work confirm that, empirically, the quality of those subsets found by the proposed technique generally outperforms those returned by the others.

VII. Conclusion

This paper has described a flexible FS method based on parameter-controlled HS. An iterative refinement technique is also presented aiming to find the quality feature subset(s) with compact size. This work offers a number of advantages over conventional approaches: fast convergence, simplicity, insensitivity to initial states, and efficiency in finding minimal subsets.

Experimental comparative studies show that the HS is capable of identifying good-quality feature subsets for most test data sets. The resulting classification accuracy tested upon discovered fuzzy-rough reducts is comparable to that using the full feature subsets. In almost all aspects, the HS approach delivered considerably better results than GAs, and it is also superior than PSO in many cases. The proposed improvements, namely parameter control and iterative refinement, further improve the HS performance, making it a strong search mechanism for data sets with large amount of features.

The proposed approach is general and can be used in conjunction with other filter-based [33] and wrapper-based [22] subset evaluation techniques [4], [5]. Owing to the underlying randomized and yet simple nature, the entire solution space of a given problem may be examined by running the HS algorithm in parallel. This will help to reveal a number of solutions much quicker than random search or exhaustive search methods. Experimental evaluation of these ideas remains an active research. Alternative approaches to t-tests may provide a means of further examining the statistical significance of the proposed research. For instance, a measurement that can study multiple algorithms in the context of a number of different data sets [12] may be adopted so that an overall conclusion of the algorithms' performances can be reached.

Although promising, much can be done to further improve the potential of the present work. For example, currently, the total number of iterations is predefined, but a good subset may be found early during the search process. It would be useful to develop a better stopping criterion based on the rate of improvement of the emerging solution and/or the overall quality of the entire harmony memory. This way, the run-time efficiency will become adaptive to the complexity of the problem itself; a further performance improvement can therefore be expected. A better iterative refinement algorithm may also be developed not only to produce good results but also to find them in lesser time, using perhaps an exponentially reduced feature selector size in relation to every refinement with backtracking. In addition, different versions of imprecise rough sets may be utilized to build the underlying theoretical foundation [35] as an alternative to fuzzy-rough sets.

Finally, it is worth noting that the HS-based techniques are capable of finding many reducts, owing to its randomized global search nature. However, more investigations are needed in utilizing the pool of reducts to build classifier ensembles [14], [55]. In addition, the subset evaluators employed in this work indicate that certain evaluators are more biased than the rest toward maintaining end classification accuracy or toward minimizing the feature subset size. Further investigations are thus necessary to analyze such behavior where an FS ensemble may be built by combining these measurements together. Working with multiple subsets rather than a single one has indeed become a strong trend in data mining. For this, the principles adopted in developing advanced compositional modeling mechanisms [17] may be useful to provide a generic framework to support the evaluation and composition of informative feature subsets.

Acknowledgment

The authors would like to thank the anonymous referees for their constructive comments which were very helpful in revising this paper.

References

[1] R. E. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.

[2] Y. Bengio and Y. Grandvalet, "Bias in estimating the variance of K-fold cross-validation," in *Statistical Modeling and Analysis for Complex Data Problems*. New York: Springer-Verlag, 2005, pp. 75–95.

[3] R. B. Bhatt and M. Gopal, "On fuzzy-rough sets approach to feature selection," *Pattern Recognit. Lett.*, vol. 26, no. 7, pp. 965–975, May 2005.

[4] T. Boongoen, C. Shang, N. Iam-On, and Q. Shen, "Extending data reliability measure to a filter approach for soft subspace clustering," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1705–1714, Dec. 2011.

[5] T. Boongoen and Q. Shen, "Nearest-neighbor guided evaluation of data reliability and its applications," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 6, pp. 1622–1633, Dec. 2010.

[6] D. Chen, C. Wang, and Q. Hu, "A new approach to attribute reduction of consistent and inconsistent covering decision systems with covering rough sets," *Inf. Sci.*, vol. 177, no. 17, pp. 3500–3518, Sep. 2007.

[7] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 89–106, Feb. 2011.

[8] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artif. Intell.*, vol. 151, no. 1/2, pp. 155–176, Dec. 2003.

[9] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, no. 3, pp. 131–156, 1997.

[10] K. Das Sharma, A. Chatterjee, and A. Rakshit, "Design of a hybrid stable adaptive fuzzy controller employing Lyapunov theory and harmony search algorithm," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 6, pp. 1140–1447, Nov. 2010.

[11] J. C. W. Debuse and V. J. Rayward-Smith, "Feature subset selection within a simulated annealing data mining algorithm," *J. Intell. Inf. Syst.*, vol. 9, no. 1, pp. 57–81, Jul./Aug. 1997.

[12] J. Damšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, no. 1, pp. 1–30, Dec. 2006.

[13] R. Diao and Q. Shen, "Two new approaches to feature selection with harmony search," in *Proc. 19th Int. Conf. Fuzzy Syst.*, 2010, pp. 3161–3167.

[14] R. Diao and Q. Shen, "Fuzzy-rough classifier ensemble selection," in *Proc. 20th Int. Conf. Fuzzy Syst.*, 2011, pp. 1516–1522.

[15] D. Dubois and H. Prade, "Putting rough sets and fuzzy sets together," in *Intelligent Decision Support*. Dordrecht, The Netherlands: Kluwer, 1992.

[16] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems," *Comput. Methods Appl. Mech. Eng.*, vol. 197, no. 33–40, pp. 3080–3091, 2008.

[17] X. Fu and Q. Shen, "Fuzzy compositional modeling," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 4, pp. 823–840, Aug. 2010.

[18] Z. W. Geem, *Recent Advances in Harmony Search Algorithm, Studies in Computational Intelligence*. Berlin, Germany: Springer-Verlag, 2010.

[19] M. A. Hall, "Correlation-based feature subset selection for machine learning," Ph.D. dissertation, Univ. Waikato, Hamilton, New Zealand, 1998.

[20] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 359–366.

[21] A. Hedar, J. Wangy, and M. Fukushima, "Tabu Search for Attribute Reduction in Rough Set Theory," *J. Soft Comput.—Fusion Found., Methodologies Appl.*, vol. 12, no. 9, pp. 909–918, Apr. 2006.

[22] C. Hsu, H. Huang, and S. Dietrich, "The ANNIGMA-wrapper approach to fast feature selection for neural nets," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 32, no. 2, pp. 207–212, Apr. 2002.

[23] R. Jensen and C. Cornelis, "A new approach to fuzzy-rough nearest neighbour classification," in *Proc. 6th Int. Conf. Rough Sets Current Trends Comput.*, 2008, pp. 310–319.

[24] R. Jensen and Q. Shen, *Computational Intelligence and Feature Selection: Rough and Fuzzy Approaches*. Piscataway, NJ: Wiley-IEEE Press, 2008.

[25] R. Jensen and Q. Shen, "New approaches to fuzzy-rough feature selection," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 824–838, Aug. 2009.

[26] G. H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proc. 11th Conf. Uncertainty Artif. Intell.*, San Mateo, CA, 1995, pp. 338–345.

[27] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy, "Improvements to Platt's SMO algorithm for SVM classifier design," *Neural Comput.*, vol. 13, no. 3, pp. 637–649, Mar. 2001.

[28] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1/2, pp. 273–324, Dec. 1997.

[29] R. Leardi, R. Boggia, and M. Terrile, *Genetic Algorithms as a Strategy for Feature Selection*, vol. 6, no. 5, pp. 267–281, Sep./Oct. 1992.

[30] H. Lee, C. Chen, J. Chen, and Y. Jou, "An efficient fuzzy classifier with feature selection based on fuzzy entropy," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 3, pp. 426–432, Jun. 2001.

[31] K. S. Lee and Z. W. Geem, "A new structural optimization method based on the harmony search algorithm," *Comput. Struct.*, vol. 82, no. 9/10, pp. 781–798, Apr. 2004.

[32] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 36–38, pp. 3902–3933, Sep. 2005.

[33] H. Liu and H. Motoda, Eds., *Computational Methods of Feature Selection*. London, U.K.: Chapman & Hall/CRC, 2008, ser. Data Mining and Knowledge Discovery.

[34] H. Liu and L. Yu, "Toward intergrating feature selection algorithms for classification and clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 4, pp. 491–502, Apr. 2005.

[35] N. Mac Parthaláin and Q. Shen, "Exploring the boundary region of tolerance rough sets for feature selection," *Pattern Recognit.*, vol. 42, no. 5, pp. 655–667, May 2009.

[36] N. Mac Parthaláin, Q. Shen, and R. Jensen, "A distance measure approach to exploring the rough set boundary region for attribute reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 3, pp. 305–317, Mar. 2010.

[37] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Appl. Math. Comput.*, vol. 188, no. 2, pp. 1567–1579, May 2007.

[38] M. Mahdavi, M. Haghir Chehreghani, H. Abolhassani, and R. Forsati, "Novel meta-heuristic algorithms for clustering web documents," *Appl. Math. Comput.*, vol. 201, no. 1/2, pp. 441–451, Jul. 2008.

[39] M. H. Mashinchi, M. A. Orgun, M. Mashinchi, and W. Pedrycz, "A tabu-harmony search-based approach to fuzzy linear regression," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 3, pp. 432–448, Jun. 2011.

[40] F. Min, H. He, Y. Qian, and W. Zhu, "Test-cost-sensitive attribute reduction," *Inf. Sci.*, vol. 181, no. 22, pp. 4928–4942, Nov. 2011.

[41] D. P. Muni, N. R. Pal, and J. Das, "Genetic programming for simultaneous feature selection and classifier design," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 36, no. 1, pp. 106–117, Feb. 2006.

[42] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, *UCI Repository of Machine Learning Databases*. Irvine, CA: Dept. Inf. Comput. Sci., Univ. California, 1998. [Online]. Available: http://www.ics.uci.edu/mlearn/MLRepository.html

[43] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data*. Dordrecht, The Netherlands: Kluwer, 1991.

[44] L. Polkowski, T. Y. Lin, and S. Tsumoto, Eds., "Rough set methods and applications: New developments in knowledge discovery in information systems," in *Studies in Fuzziness and Soft Computing*. New York: Physica-Verlag, 2000.

[45] C. C. O. Ramos, A. N. Souza, G. Chiachia, A. X. Falcão, and J. P. Papa, "A novel algorithm for feature selection using Harmony Search and its application for non-technical losses detection," *Comput. Elect. Eng.*, vol. 37, no. 6, pp. 886–894, Nov. 2011.

[46] M. Shah, M. Marchand, and J. Corbeil, "Feature selection with conjunction of decision stumps and learning from microarray data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 174–186, Jan. 2012.

[47] Q. Shen and A. Chouchoulas, "A rough-fuzzy approach for generating classification rules," *Pattern Recognit.*, vol. 35, no. 11, pp. 2425–2438, Nov. 2002.

[48] D. Slezak and S. Widz, "Rough-set-inspired feature subset selection, classifier construction, and rule aggregation," in *Proc. 6th Int. Conf. Rough Sets Knowl. Technol.*, 2011, pp. 81–88.

[49] R. Srinivasa Rao, S. V. L. Narasimham, M. Ramalinga Raju, and A. Srinivasa Rao, "Optimal network reconfiguration of large-scale distribution system using harmony search algorithm," *IEEE Trans. Power Syst.*, vol. 26, no. 3, pp. 1080–1088, Aug. 2011.

[50] R. W. Swiniarski and A. Skowron, "Rough set methods in feature selection and recognition," *Pattern Recognit. Lett.*, vol. 24, no. 6, pp. 833–849, Mar. 2003.

[51] E. C. C. Tsang, D. Chen, D. S. Yeung, X. Wang, and J. W. T. Lee, "Attributes reduction using fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 16, no. 5, pp. 1130–1141, Oct. 2008.

[52] A. Vasebi, M. Fesanghary, and S. M. T. Bathaee, "Combined heat and power economic dispatch by harmony search algorithm," *Int. J. Elect. Power Energy Syst.*, vol. 29, no. 10, pp. 713–719, Dec. 2007.

[53] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognit. Lett.*, vol. 28, no. 4, pp. 459–471, Mar. 2007.

[54] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco, CA: Morgan Kaufmann, 2005.

[55] J. Wróblewski, "Ensembles of classifiers based on approximate reducts," *Fundamenta Informaticae*, vol. 47, no. 3/4, pp. 351–360, Oct. 2001.

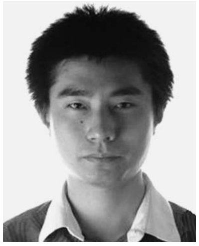
[56] J. Wróblewski, "Finding minimal reducts using genetic algorithm," in *Proc. 2nd Annu. Join Conf. Inf. Sci.*, 1995, pp. 186–189.

[57] E. P. Xing, M. I. Jordan, and R. M. Karp, "Feature selection for high-dimensional genomic microarray data," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 601–608.

[58] Y. Yao and R. Fu, "Partitions, coverings, reducts and rule learning in rough set theory," in *Proc. 6th Int. Conf. Rough Sets Knowl. Technol.*, 2011, pp. 101–109.

[59] R. Zhang and L. Hanzo, "Iterative multiuser detection and channel decoding for DS-CDMA using harmony search," *Signal Process. Lett.*, vol. 16, no. 10, pp. 917–920, Oct. 2009.

[60] Z. Zhu, Y. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, no. 1, pp. 70–76, Feb. 2007.



Ren Diao received the B.A. and M.A. degrees in computer science from the University of Cambridge, Cambridge, U.K. He is currently working toward the Ph.D. degree in the Department of Computer Science, Aberystwyth University, Aberystwyth, U.K.

He is currently a Member of the Advanced Reasoning Group, Department of Computer Science. His research interests include fuzzy set theory, nature-inspired heuristics, and machine learning.

Qiang Shen received the Ph.D. degree from Heriot-Watt University, Edinburgh, U.K.

He holds the established chair in computer science and is currently the Head of the Department of Computer Science, Aberystwyth University, Aberystwyth, U.K. He is a Fellow of the Learned Society of Wales. His research interests include computational intelligence, fuzzy and qualitative modeling, reasoning under uncertainty, pattern recognition, data mining, and real-world applications of such techniques for intelligent decision support (e.g., crime detection, consumer profiling, systems monitoring, and medical diagnosis).

Prof. Shen is currently an Associate Editor of two premier IEEE TRANSACTIONS (IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS–PART B: CYBERNETICS and IEEE TRANSACTIONS ON FUZZY SYSTEMS) and an editorial board member of several other leading international journals. He has authored 2 research monographs and around 300 peer-reviewed papers, including one which received an Outstanding Transactions Paper Award from IEEE.