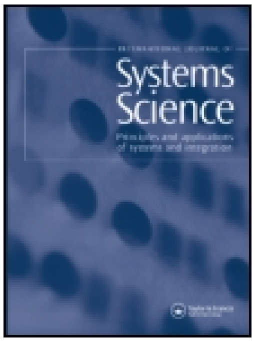


International Journal of Systems Science

ISSN: 0020-7721 (Print) 1464-5319 (Online) Journal homepage: http://www.tandfonline.com/loi/tsys20

Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm

Salam Salameh Shreem, Salwani Abdullah & Mohd Zakree Ahmad Nazri

To cite this article: Salam Salameh Shreem, Salwani Abdullah & Mohd Zakree Ahmad Nazri (2016) Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm, International Journal of Systems Science, 47:6, 1312-1329, DOI: 10.1080/00207721.2014.924600

To link to this article: http://dx.doi.org/10.1080/00207721.2014.924600

Published online: 06 Jun 2014.

International Journal of Systems Science, 2016
Vol. 47, No. 6, 1312–1329, http://dx.doi.org/10.1080/00207721.2014.924600

Hybrid feature selection algorithm using symmetrical uncertainty and a harmony search algorithm

Salam Salameh Shreem[†], Salwani Abdullah[*] and Mohd Zakree Ahmad Nazri

Data Mining and Optimisation Research Group (DMO), Centre for Artificial Intelligent (CAIT), Universiti Kebangsaan Malaysia, Bangi Selangor, Malaysia

(*Received 8 April 2013; accepted 4 May 2014*)

Microarray technology can be used as an efficient diagnostic system to recognise diseases such as tumours or to discriminate between different types of cancers in normal tissues. This technology has received increasing attention from the bioinformatics community because of its potential in designing powerful decision-making tools for cancer diagnosis. However, the presence of thousands or tens of thousands of genes affects the predictive accuracy of this technology from the perspective of classification. Thus, a key issue in microarray data is identifying or selecting the smallest possible set of genes from the input data that can achieve good predictive accuracy for classification. In this work, we propose a two-stage selection algorithm for gene selection problems in microarray data-sets called the symmetrical uncertainty filter and harmony search algorithm wrapper (SU-HSA). Experimental results show that the SU-HSA is better than HSA in isolation for all data-sets in terms of the accuracy and achieves a lower number of genes on 6 out of 10 instances. Furthermore, the comparison with state-of-the-art methods shows that our proposed approach is able to obtain 5 (out of 10) new best results in terms of the number of selected genes and competitive results in terms of the classification accuracy.

Keywords: feature selection; filter; harmony search; microarray; wrapper

1. Introduction

The main purpose of feature selection for microarray data is to construct a model that can distinguish genes efficiently, that is, to find a small set of genes that can provide high classification accuracy (Chuang, Yang, Wu, & Yang, 2011). In recent years, many researchers have conducted studies on gene selection. With the rapid progress in database technologies today, data-sets with a very large number of features or genes are ubiquitous in data mining and machine learning applications, as well as in many academic communities. Microarray data are characterised by very large numbers of genes but a small number of samples. Thus, learning from microarray data is an arduous task because machine learning only works well on small numbers of genes (Agrawal & Bala, 2007). Gene selection addresses this problem by eliminating the redundant and irrelevant genes.

Gene selection for microarray data-sets is challenging because of the considerably large number of genes in comparison to the limited number of samples. In this case, most of the genes are irrelevant (not related to the given cancer classification) and redundant. Therefore, a set of informative genes must be selected to improve classification accuracy.

Given an input data with N genes, gene selection is generally carried out to identify the smallest possible set of genes among the competing subsets of genes that can maximise the classification accuracy. Gene selection problems can be formalised as a combinatorial optimisation problem in which the search space is a set of all possible subsets (Duval & Hao, 2010; Liu & Motoda, 1998). This problem is known as NP-hard (Amaldi & Kann, 1998) and is a highly combinatorial search problem. The search space increases exponentially when the number of genes increases, and 2^N subsets of genes are possible, where N represents the number of genes. Optimisation methods such as metaheuristics and their hybridisation have been successfully applied to solve gene selection problems because these methods can maintain a population of solutions that can handle a problem with a complex and very large search space (Lozano & García-Martínez, 2010) (e.g. genetic algorithm (GA; Agrawal & Bala, 2007; Chuang et al., 2011; El Akadi, Amine, El Ouardighi, & Aboutajdine, 2011), particle swarm optimisation (PSO; Talbi, Jourdan, Garcia-Nieto, & Alba, 2008), hybrid GA and a binary PSO (Chuang & Yang, 2009; Chuang, Yang, & Li, 2011), tabu search (Chuang & Yang, 2009; Zhang & Sun, 2002), memetic algorithm (MA; Zhu, Ong, & Dash, 2007a), and ant colony optimisation (ACO; Huang, 2009).

The gene selection models in the literature can be grouped into two main models: filter models and

[*] Corresponding author. Email: salwani@ukm.edu.my

[†] Present address: Saudi Electronic University, Riyadh, Saudi Arabia.

wrapper models. The filter model is utilised to select a subset of genes before applying the classification process; the genes are ranked according to their individual relevance or discriminative power before the learning algorithm is applied with regard to the target classes (Dash & Liu, 1997). However, a filter approach does not account for the interactions among the features. By contrast, the wrapper models usually utilise machine learning techniques as a fitness function to measure the feature subsets. The wrapper method performs better than filter models but is usually computationally more expensive than the filter method (Kohavi & John, 1997). Related literature summarises and suggests that the filter and wrapper methods are complementary. The hybridisation between them yields better performance compared with a filter or wrapper method alone. Therefore, the success of the above-mentioned hybrid methods in literature is our main motivation for proposing a new hybrid filter wrapper method for gene selection problems. This proposed method employs symmetrical uncertainty (SU) as a filter method and a harmony search algorithm (HSA) as a wrapper method.

Despite the variety of meta-heuristic methods for gene selection that currently exist (Agrawal & Bala, 2007; Chuang & Yang, 2009; Duval & Hao, 2010; El Akadi et al., 2011), a solution method that can enhance the current results and address problems that have a very large number of genes is still needed. The success of the hybrid population-based methods in solving gene selection problems and other complex problems (Lozano & García-Martínez, 2010; Zhu et al., 2007a) has motivated us to propose a new population-based method that combines SU with an HSA. SU is used to help remove the irrelevant genes, and HSA is chosen for the gene selection problem because it is a stochastic random search method that is less parameter sensitive and overcomes the drawback of the building block theory of GAs by considering all of the existing solutions instead of considering only two solutions (parents) in its reproduction (Geem, Kim, & Loganathan, 2001; Mahdavi, Fesanghary, & Damangir, 2007). Our aim is to propose a new population-based meta-heuristic method for the gene selection problem that hybridises SU and the HSA. The proposed method, hereafter referred to as the SU-HSA, can handle a very large number of genes and enhance classification accuracy using two different classifiers, namely, IB1 and Naïve Bayes (NB).

The performance of the SU-HSA is assessed using well-known challenging microarray data-sets (Li, Zhang, & Ogihara, 2004). We choose these data-sets for the following reasons. First, these data-sets represent real-world applications and pose a substantial challenge to the research community because of a very large number of genes and a limited number of samples. Second, the state-of-the-art results can still be improved.

We perform comprehensive experiments to investigate significant differences between a hybrid method (SU-HSA) and the HSA alone. The results demonstrate that in terms of both the number of selected features and the classification accuracy, the SU-HSA is more competitive (if not better in some cases) compared with the HSA alone and with the best-known methods in literature.

The remainder of this paper is organised as follows. Section 2 presents a set of well-known related studies. Section 3 outlines the proposed SU-HSA. Section 4 discusses the experimental results. Section 5 concludes the study.

2. Hybrid symmetrical uncertainty and reference set harmony search algorithm

Our proposed method is a hybrid filter–wrapper that treats SU as a filter and an HSA as a wrapper. This hybridised algorithm is called SU-HSA. The components and process are discussed in the following subsections.

2.1. First stage: symmetrical uncertainty

SU is a simple and efficient procedure to evaluate the goodness of genes for classifying the genes and the target concepts; it has been utilised by many researchers to evaluate the goodness of features (Bermejo, Gámez, & Puerta., 2011; Chuang, Yang, Wu, & Yang, 2011b; Hall, 1999; Senthamarai Kannan & Ramaraj, 2010; Sun et al., 2012). SU is thus chosen in the present study. The SU value is calculated for each gene, and the genes are ranked in descending order according to their SU values. In general, the genes with the highest SU value have a high probability to be used to guide the initialisation of the harmony memory (HM) for HSA in the next stage. The genes with a low SU value are removed.

The SU correlation measure is based on the information-theoretical concept of entropy (Senthamarai Kannan & Ramaraj, 2010), which is a measure of the uncertainty of a random variable. The entropy of a variable X is defined as follows:

$$H(X) = -\sum\nolimits_i P(x_i)\log_2(P(x_i)) \tag{1}$$

and the entropy of X after observing values of another variable Y is defined as follows:

$$H(X|Y) = -\sum\nolimits_j P(x_i)\sum\nolimits_i (x_i|y_j)\log_2(x_i|y_j)) \tag{2}$$

where $P\,(x_i)$ is the prior probability for all of the values of X, and $P\,(x_i|y_i)$ is the posterior probabilities of X given the values of Y. The amount by which the entropy of X decreases reflects additional information about X that is provided by Y, and is called the information gain (IG); IG is given by the following equation:

$$\mathrm{IG}(X|Y) = H(X) - H(X|Y) \tag{3}$$

According to this measure, a feature Y is regarded as more correlated to feature X than to feature Z, if $\text{IG}(X|Y) > \text{IG}(Z|Y)$.

Information gain is symmetrical for two random variables, X and Y. Symmetry is a property that measures the correlation between features. However, IG is biased in favour of features with more values. Values must be normalised to ensure that they are comparable and have the same effect. Therefore, we choose SU (Hall, 1999), which is defined as follows:

$$\text{SU} = 2.0 \times \left[\frac{\text{IG}(X|Y)}{H(X) + H(Y)} \right] \tag{4}$$

SU compensates for the IG's bias towards features with more values. These values are normalised within the range [0, 1]. The value 1 indicates that knowledge of either one of the values completely predicts the value of the other, whereas the value 0 indicates that X and Y are independent. SU treats a pair of features symmetrically.

2.2. Second stage: harmony search algorithm

The wrapper approach is used in gene selection problems (El Akadi et al., 2011) to distinguish among tumour types, minimise the number of genes, and assist in drug discovery and early diagnosis.

The wrapper in the gene selection problem can be considered as a combinatorial search problem because the search space expands exponentially with the number of genes (Duval, Hao, & Hernandez Hernandez, 2009). Finding the optimal solution by exploring the whole search space is thus very difficult because exploring the whole search space to locate the local optimum takes a long time. A solution space normally has a large number of local optimal solutions that usually cannot be tackled using classical methods (Lin & Chen, 2012). Thus, meta-heuristic techniques would be a convenient way to obtain a good solution without exploring the whole space of solutions (Yusta, 2009). In the current paper, we use the HSA as a wrapper approach.

The HSA was proposed by Geem et al. (2001) and is one of the most recently developed population-based meta-heuristic optimisation techniques; it has been applied successfully in many optimisation problems, such as in structural design (Geem, 2009a; Lee & Geem, 2004), energy system dispatch (Vasebi, Fesanghary, & Bathaee, 2007), music composition, (Geem & Choi, 2007) sudoku puzzle solving (Geem, 2007), webpage clustering (Mahdavi, Chehreghani, Abolhassani, & Forsati, 2008), soil stability analysis (Cheng, Li, Lansivaara, Chi, & Sun, 2008), ground water modelling (Ayvaz, 2007; Tamer Ayvaz, 2009), heat exchanger design (Fesanghary, Damangir, & Soleimani, 2009), medical physics (Panchal, 2009), medical image analysis (Alia, Mandava, & Aziz, 2010; Alia, Mandava, Ramachandram, & Aziz, 2009a), timetabling (Al-Betar & Khader, 2008; Al-Betar, Khader, & Liao, 2010), image segmentation (Alia, Mandava, Ramachandram, & Aziz, 2009b), manufacturing optimisation problems (Yildiz, 2012), and document clustering (Forsati, Mahdavi, Shamsfard, & Reza Meybodi, 2013). Other applications of HSA can be found in El-Abd (2012), Hasan, Abu Doush, Al Maghayreh, Alkhateeb, and Hamdan (2014), Lee and Mun (2014), Maheri and Narimani (2014), Nekooei, Farsangi, Nezamabadi-Pour, and Lee (2013), Sirjani, Mohamed, and Shareef (2012), and Wang et al. (2013). HSA has the following advantages:

(1) It has fewer mathematical requirements and does not require initial value settings of the decision variables (Mahdavi et al., 2007).
(2) It generates a new solution after considering all of the existing vectors.
(3) It can explore the search space in a parallel optimisation environment, in which each solution (harmony) vector is generated by intelligently exploring and exploiting a search space (Geem, 2009b).

These advantageous features make the HSA very flexible and capable of producing improved solutions.

The HSA imitates the musical performance process that occurs when a musician attempts to find a state of harmony. In the music production process, a musician selects and collects different notes from a whole range of notes. The musician then plays the notes with musical instruments to search for an ideal state of harmony in the music. This kind of process is similar to an optimum design process that involves the search for an optimum solution for optimisation problems; the solution is determined by the objective function. The pitch of each musical instrument determines the aesthetic quality, just as the fitness function or objective function is determined by the set of values assigned to each design variable. Aesthetic sound quality can be iteratively improved the same way a fitness function value can be improved iteration after iteration (Geem et al., 2001).

During the improvisation process of music, a musician has three possible options: (1) play any famous piece of music from memory, (2) play notes that are almost similar to a known piece of music (adjusting the pitch slightly), or (3) compose new or random notes. Geem et al. (2001) formalised these three choices into a quantitative optimisation process. The three corresponding components are HM, pitch adjustment, and randomisation.

Figure 1 illustrates the analogy between music improvisation and optimisation problems. Consider a jazz trio composed of a cello, a guitar, and a saxophone. The memory of each musician has a number of preferable pitches: cello = {La, Do, Re}, guitar = {Do, Re, Fa}, and saxophone = {Mi, Sol, La}; these pitches correspond to the variable values $x_1 = \{3.0, 2.2, 4.0\}$, $x_2 = \{1.0, 3.3, 5.0\}$,

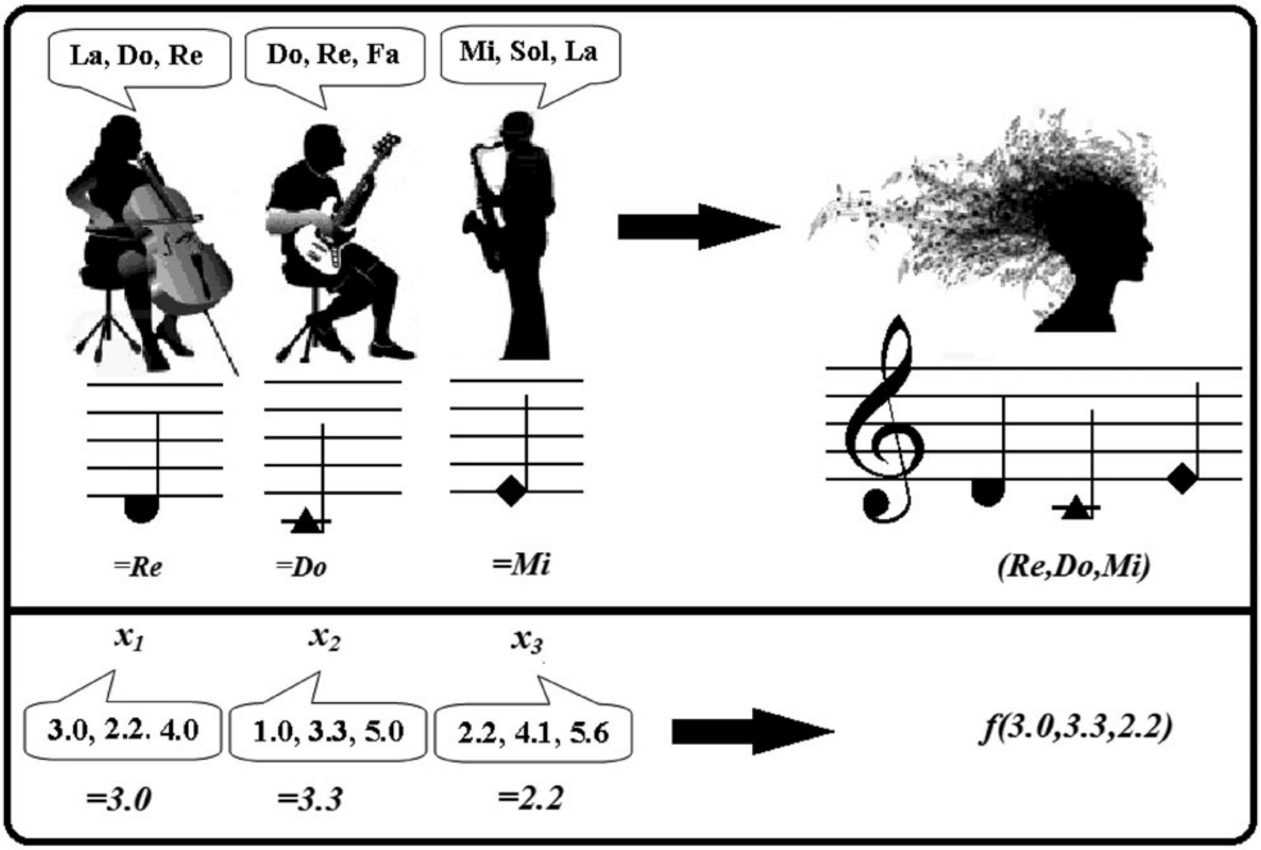


Figure 1. Analogy between music improvisation and optimisation.

and $x_3 = \{2.2, 4.1, 5.6\}$. If the cellist randomly plays the note {Re} from {La, Do, Re}, the guitarist plucks the note {Do} from {Do, Re, Fa}, and the saxophonist plays the note {Mi} from {Mi, Sol, La}, they make a new harmony {Re, Do, Mi}. If the new harmony is better than the existing poor harmony in the HM, the new harmony is kept while the poor harmony is excluded from the HM. This procedure is repeated until the perfect harmony is found. Assuming the selected values for the new solution vector (3.0, 3.3, 2.2), this solution is evaluated by the objective function. If the objective function is better than the existing poor harmony in the HM, it is retained. Otherwise, it is excluded.

Some existing equivalences between the musical terms and optimisation based on the above description are illustrated in Table 1.Each instrument or musician corresponds to each decision variable, the pitch of each musical instrument corresponds to the value of a decision variable, a new harmony produced by all the musical instruments corresponds to a solution for the optimisation problems, the aesthetic quality of a harmony corresponds to the objective function value of a solution, and the musical harmony iteratively improving practice after practice corresponds to enhancing the solution vector iteration after iteration.

Table 1. Optimisation terms in the musical context.

Musical terms	Optimisation terms
Improvisation	Generation
Harmony	Solution vector
Musician	Decision variable
Pitch	Value
Pitch range	Value range
Audio-aesthetic standard	Objective function
Practice	Iteration
Pleasing harmony	Optimal solution

As shown in Figure 9, the HSA consists of the following steps:

Step 1: Initialise the problem and the algorithm parameters.
Step 2: Initialise the HM.
Step 3: Improvise the new harmony (x').
Step 4: Update the HM.
Step 5: Check the stopping criterion.
Step 6: Cadenza.

These steps are described in the next subsections.

$$HM\ matrix = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_N^1 & \rightarrow & f(X^1) \\ x_1^2 & x_2^2 & \dots & x_N^2 & \rightarrow & f(X^2) \\ \vdots & \vdots & \dots & \vdots & \rightarrow & \vdots \\ x_1^{HMS-1} & x_2^{HMS-1} & \dots & x_N^{HMS-1} & \rightarrow & f(X^{HMS-1}) \\ x_1^{HMS} & x_2^{HMS} & \dots & x_N^{HMS} & \rightarrow & f(X^{HMS}) \end{bmatrix}$$

Figure 2. HM as a matrix of solutions.

Step 1. Initialise the problem and the algorithm parameters

In this step, the HSA parameters are initialised. The parameters comprise the following:

- Harmony memory size (HMS): HMS defines the number of harmonies (solutions) that are stored in the HM that is similar to the population size in the GA.
- Harmony memory consideration rate (HMCR): HMCR is used during the improvisation process to decide whether the value of a decision variable of a new harmony should be chosen from the value of any harmony in the HM or at random from the possible range space. HMCR usually takes a value in the range [0, 1]. The probability of randomly selecting the value of the decision variable from the possible range is given as 1 − HMCR.
- Pitch adjusting rate (PAR): PAR is also used during the improvisation process to determine whether to modify the values of decision variables that have been selected from the HM to its neighbouring value or to make no change. PAR takes a value in the range [0, 1].
- Number of improvisations (NI): This parameter represents the number of times that the HSA is repeated; it is treated as a stopping criterion.

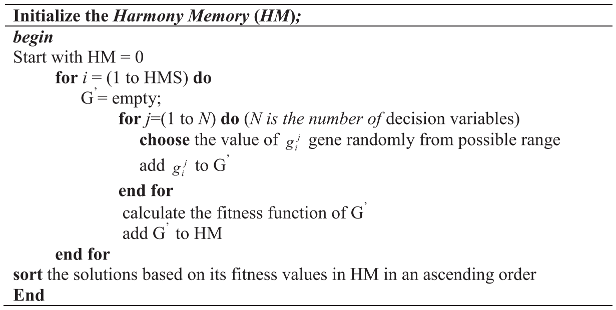
Step 2. Initialise the HM

HM is a two-dimensional matrix of solutions that has a size equal to the HMS. Each row in the HM represents one chromosome (solution), as shown in Figure 2. The solutions are randomly generated with their respective values of the fitness function $f(x')$. Solutions in the HM are arranged in a reverse order based on the values of the fitness function.

Each chromosome is composed of a bit string that has a length equal to the total number of genes that encode a candidate feature subset. A bit of 1 (0) implies that the corresponding feature is selected (excluded). For example, $X' = [1100100011 \rightarrow 60]$ represents a solution (chromosome) X' that contains genes 1, 2, 5, 9, and 10; the quality of the solution is 60. The fitness function is utilised to calculate the fitness value for each chromosome in the HM; it is obtained by using the following induction algorithm:

$$f(X) = J(S_x) \tag{5}$$

where S_x denotes the selected gene subset (i.e. 1, 2, 5, 9, and 10 as in the above example) that is encoded in a chromosome X, and $J(S_x)$ is the generalisation error obtained for the selected genes (S_x) that can be estimated using cross-validation (CV) with the classification algorithm (e.g. NB and IB1). The pseudo-code of the HM initialisation is presented in Figure 3.



```
Initialize the Harmony Memory (HM);
begin
Start with HM = 0
    for i = (1 to HMS) do
        G` = empty;
            for j=(1 to N) do (N is the number of decision variables)
                choose the value of $g_i^j$ gene randomly from possible range
                add $g_i^j$ to G`
            end for
            calculate the fitness function of G`
            add G` to HM
    end for
sort the solutions based on its fitness values in HM in an ascending order
End
```

Figure 3. Pseudo-code for initialising the harmony memory. Note: HMS is the size of the HM and N is the length of harmony vectors.

Step 3. Improvise the new harmony

Generating a new harmony (which is the core of the HSA) is called 'improvisation.' The HSA improvises (generates) a new harmony $X' = x_1', x_2', \dots, x_N'$ using three rules: (1) memory consideration, (2) pitch adjustment, and (3) random consideration.

(1) *Memory consideration*

In memory consideration, the value of the first gene x_1' for the new solution is selected from any existing values that are stored in the HM ($x_1', \dots, x_1^{\mathrm{HMS}}$) with the HMCR probability. The next gene x_2' is also chosen from ($x_2', \dots, x_2^{\mathrm{HMS}}$), and the values for the other genes are selected in the same manner if they satisfy the probability selection of HMCR between 0 and 1. Assume that the HMCR is equal to 0.7. This condition means that when the HSA generates a new solution during the improvisation process, the HSA performs the following: a random number between 0 and 1 is generated. If the generated random number is less than 0.7, then the gene value is selected from the HM ($x_1', \dots, x_1^{\mathrm{HMS}}$). If the generated random number is greater than 0.7, the value of the gene is determined based on the random consideration process. In random consideration, the other genes that are not selected from the memory with the probability of 1–HMCR are selected randomly according to their possible range of values (as shown in Equation (6)). This condition means that the diversity increases, and that various

```
Improvise a new harmony
Begin
for $i$ = 1 to the maximum number of iterations (NI)
    $X'$ = empty
    for $j$ = 1 to number of decision variables
        $r1$ = uniform random number between [0,1]
            if (r1 < HMCR)
                $x^{i}_{j}$ = randomly selected from HM
                $r2$ = uniform random number between [0,1]
                if (r2 <PAR)
                    $mutatex^{j}_{i}$
                  Else
                    do not change the decision variable value
                  end if
            Else
                $x^{j}_{i}$ = randomly generated from the possible range of the
                considered decision variable (with probability 1-HMCR)
            end if
    end for $j$
    update HM
end for $i$
```

Figure 4. Improvisation of a new harmony.

solutions can be explored to obtain the global optimum:

$$x'_i \leftarrow \begin{cases} x'_i \in \{x_i^1, x_i^2, \ldots, x_i^{\mathrm{HMS}}\} & \text{w.p} \quad \mathrm{HMCR} \\ x'_i \in X^i & \text{w.p} \quad (1-\mathrm{HMCR}) \end{cases} \tag{6}$$

For example, if the HMCR = 0.70, then the genes in the new solution are chosen from the HM with a probability of 70% (memory consideration) and from the possible range of values with a 30% probability (random consideration). These cumulative steps ensure that good harmonies are considered as the elements of a new harmony.

(2) *Pitch adjustment*

Every gene obtained by the HMCR is examined to determine whether it should be tuned (pitch-adjusted) with the probability of the PAR or left unchanged with the probability 1 – PAR. The adjustment involves the mutation of the gene from 0 to 1 or from 1 to 0. This operation uses the PAR as follows:

$$X'_i \leftarrow \begin{cases} \text{mutate}\, x'_1 & \text{w.p} \quad \mathrm{PAR} \\ x'_1 & \text{w.p} \quad 1-\mathrm{PAR} \end{cases} \tag{7}$$

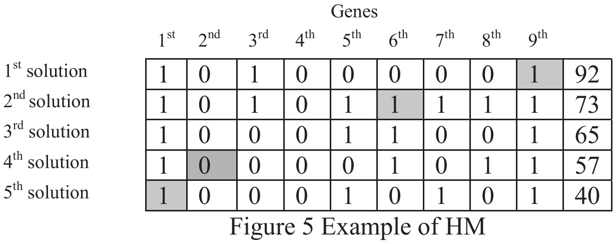
The pseudo-code of the improvisation process is given in Figure 4.

Step 4. Update the HM

To update the HM with the improvised solution$X' = [x'_1, x'_2, \ldots, x'_N]$, the fitness function is calculated for the solution. If the fitness function for the new solution is better than the worst solution in the HM, then the worst solution is excluded and replaced by the new solution. Otherwise, the new solution is ignored.

Step 5. Check the stopping criterion

The stopping criterion in this work is the maximum number of iterations or an accuracy of 100%. Otherwise, Steps 3 and 4 are repeated until the stopping criterion is reached.



	Genes									
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	
1st solution	1	0	1	0	0	0	0	0	1	92
2nd solution	1	0	1	0	1	1	1	1	1	73
3rd solution	1	0	0	0	1	1	0	0	1	65
4th solution	1	0	0	0	0	1	0	1	1	57
5th solution	1	0	0	0	1	0	1	0	1	40

Figure 5 Example of HM

Figure 5. Example of HM.

Step 6. Cadenza

A cadenza is a musical passage that occurs in the final step of a movement in a musical work to return the most fantastic harmony played during the improvisation process. In the context of the HSA, the cadenza can be referred to as the last step in the HSA that occurs at the end of the search process for the best harmony. In this process, the HSA returns the best harmony found and stored in the HM based on the fitness function *f*(*X*).

We include a numerical example on HSA as illustrated below:

Step 1. Initialise the problem and the algorithm parameters

Assume that HMS = 5; HMCR = 0.7, PAR = 0.3, NI = 50.

Step 2. Initialise the HM

There are five solutions in the HM with the length size equals to the number of genes (9 in this example, where each gene is represented by 1 or 0). The quality of each solution is shown in the last column of Figure 5. For example, the first solution contains genes 1, 3, and 9; the quality of the solution is 92.

Step 3. Improvise the new harmony (x')

Assume that the generated random numbers ($r1$) are 0.2, 0.1, 0.8, 0.9, 0.9, 0.6, 0.8, 0.9, and 0.5, which are equal to the number of genes. The generated random number will be compared with the HMCR. If it is less than HMCR, then a gene from the solution in the HM is selected. Otherwise a random consideration is utilised where a random number either 1 or 0 is chosen. A new solution is generated as follows:

(1) *Memory consideration*

First, generate an empty solution with size equals to the number of genes. Then, loop through the solution cell one by one and assign a value to each cell based on the HMCR as follows:

- The first and the second generated numbers (0.2 and 0.1, respectively) are less than HMCR (0.7), then the first and second genes are selected from the first and second columns from the HM. Assume that the

Genes									
1st	2nd	3rd	4th	5th	6th	7th	8th	9th	
1	0	0	1	0	1	1	0	1	

Figure 6. Example of the new solution after the memory consideration.

Genes									
1st	2nd	3rd	4th	5th	6th	7th	8th	9th	
0	1	0	1	0	1	1	0	0	62

Figure 7. Example of the new solution after the pitch adjustment.

selected gene is from the fourth and fifth solutions (fourth and fifth rows) which are ‘1’ and ‘0’, respectively. Note that, the highlighted text in Figure 5 represents the genes which are selected based on the memory consideration. The third generated number (0.8) is greater than HMCR (0.7). In this case, a random number either 1 or 0 is generated. Assume that 0 is generated. Thus, the third gene in the new solution will be 0. The process is repeated for all genes to form a new solution. Assume that the new solution is represented as in Figure 6. (Again, the highlighted gene represents the value that is based on the memory consideration (which are genes 1, 2, 6, and 9).)

(2) *Pitch adjustment*

For every gene that is based on the memory consideration will be tuned (pitch-adjusted) or mutated from 0 to 1 or from 1 to 0 based on the comparison between the random generated number [0,1] and PAR (note that, selected genes that are based on the random consideration will not be tuned). Assume that the generated random numbers (*r*2) are 0.2, 0.1, 0.6, and 0.2.

- The first *r*2 (0.2) is less than PAR (0.3), then the first gene from Figure 6 is mutated (changed from 1 to 0).
- The second *r*2 (0.1) is less than PAR (0.3), then the second gene from Figure 6 is mutated (changed from 0 to 1).
- The third *r*2 (0.6) is greater than PAR (0,3), then sixth gene from Figure 6 is maintained.
- The fourth *r*2 (0.2) is less than PAR (0.3), then the ninth gene from Figure 6 is mutated (changed from 1 to 0). Thus, the new solution after the pitch adjustment takes place will be as presented in Figure 7. Assume that the calculated quality of the solution is 62.

Step 4. Update the HM

The new solution (as in Figure 7) with the quality equals to 62 will replace the worst solution in the HM. So, the fifth solution in Figure 5 will be removed and it will be replaced by the new solution. Thus, the HM will be updated as presented in Figure 8.

	Genes									
	1st	2nd	3rd	4th	5th	6th	7th	8th	9th	
1st solution	1	0	1	0	0	0	0	0	1	92
2nd solution	1	0	1	0	1	1	1	1	1	73
3rd solution	1	0	0	0	1	1	0	0	1	65
4th solution	0	1	0	1	0	1	1	0	0	62
5th solution	1	0	0	0	0	1	0	1	1	57

Figure 8. Updated HM.

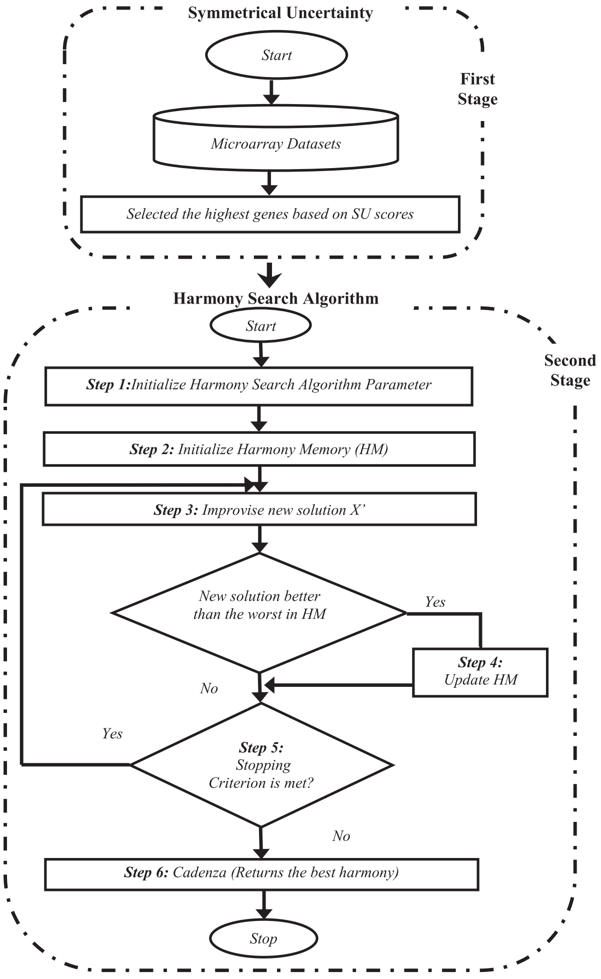

Symmetrical Uncertainty
Start
First Stage
Microarray Datasets
Selected the highest genes based on SU scores
Harmony Search Algorithm
Start
Second Stage
Step 1:Initialize Harmony Search Algorithm Parameter
Step 2: Initialize Harmony Memory (HM)
Step 3: Improvise new solution X'
New solution better than the worst in HM
Yes
Step 4: Update HM
No
Yes
Step 5: Stopping Criterion is met?
No
Step 6: Cadenza (Returns the best harmony)
Stop

Figure 9. Hybrid SU-HSA.

Step 5. Check the stopping criterion

The above steps are repeated until the NI equals to 50.

Step 6. Cadenza

HSA returns the best found solution.

2.3. *The proposed method (hybrid symmetrical uncertainty and harmony search algorithm)*

The hybrid SU-HSA is shown in Figure 9. At the beginning of the search process, the microarray data-sets are

Table 2. Description of the data-sets.

Data-set	Genes	Samples	Classes	Description
ALL–AML	7129	72	2	Two acute forms of leukaemia, i.e. acute myelogenous leukaemia (AML) and acute lymphoblastic leukaemia (ALL)
ALL–AML-3C	7129	72	3	AML, ALL B-cell, and ALL T-cell
ALL–AMl-4C	7129	72	4	AML-bone marrow, AML-peripheral blood, ALL B-cell, and T-cell.
Colon	2000	62	2	40 colon cancer biopsies versus 22 normal biopsies
CNS	7129	60	2	Outcome of the treatments for 60 central nervous system (CNS) cancer patients (21 survivors and 39 failures)
Lymphoma	4026	62	3	Three most prevalent adult lymphoid tumours
MLL	12,582	72	3	AML, ALL, and mixed-lineage leukaemia (MLL)
Breast	24,481	97	2	97 samples from breast cancer patients (46 patients developed distance metastases; the other 51 remained healthy after their initial diagnosis for an interval of at least five years)
Ovarian	15,154	253	2	The proteomic spectra of 91 normal persons and 162 ovarian cancer patients
SRBCT	2308	83	4	Small, round, blue cell tumours from childhood

pre-processed by an SU filter to eliminate redundant and irrelevant genes in the microarray data-sets. Each gene is evaluated individually and assigned a score that represents the correlation of each gene with the class. The genes are then ranked (from the highest to the lowest) based on the SU score, and the top-ranking genes are selected for use in the next selection stage (the wrapper stage).

In the second stage, the wrapper approach combines the HSA and a given classifier to accomplish the gene subset selection for the most important and discriminating genes. At the beginning of the search process in the second stage, all the parameters used in the HSA are initialised (Step 1). The HM is then initialised by randomly generating a number of initial solutions based on the highest selected genes obtained from the first stage; this number equates to the HMS (Step 2). The quality of each solution in the HM is measured based on the classification accuracy using the NB or IB1 classifiers. A new harmony or solution $X' = (x'_1, x'_2, \dots, x'_N)$ is improvised based on three rules: memory consideration, pitch adjustment, and random consideration (Step 3). If the fitness function for the new solution is better than the worst solution in the HM, then the worst solution is excluded and replaced by the new solution. Otherwise, the new solution is ignored (Step 4). This process is repeated until the stopping criterion (which is set to be either a maximum number of iterations or the classification accuracy equal to 100%) is met (Step 5). Finally, the HSA returns the best harmony found and stored in the HM based on the fitness function $f(X)$ (Step 6). This process is illustrated in Figure 9.

3. Experimental results

The proposed algorithms are programmed using Java and performed on an Intel Core i5-2450M–2.5 GHz CPU with 4 GB of RAM. The Waikato Environment for Knowledge Analysis (WEKA) is used as a classifier tool; two classifiers (IB1 and NB) are used in this study. To assess the benefit of incorporating SU in the HSA, we perform two sets of experiments. The first experiment compares the performance of the HSA with SU (denoted as SU-HSA) with that of HSA without SU (denoted as HSA) using the same parameter values and computer resources. The second experiment analyses and compares the performance of the SU-HSA with that of state-of-the-art algorithms. For both experimental tests, we report over 10 independent runs with different random seeds, the average number of selected genes, and the average of the classification accuracy rate using two different classifiers (i.e. NB and IB1) with 10-fold CV (Ambroise & McLachlan, 2002).

3.1. Data-sets

To evaluate the usefulness of the SU-HSA approach, this experiment is performed on 10 microarray data-sets. The characteristics and a brief description of the data-sets used in the experiments are summarised in Table 2, which can be downloaded at http://csse.szu.edu.cn/staff/zhuzx/Datasets.html.

3.2. Parameter settings

In the first stage of the proposed method, the SU filter stops after selecting the top one hundred genes, as recommended in El Akadi et al. (2011).

In the second stage, the NB and IB1 classifiers with 10-fold CV are applied to validate and assess the generated solutions. The 10-fold CV means that the data-sets are partitioned into training sets (90%) and independent test sets (10%). In other words, feature subsets are selected from 90% of the training instances, and then the accuracy is estimated over the unseen 10% of the test instances. The accuracy and number of the selected genes reported in this study are based on 10-fold CV and test data.

Table 3. Parameter settings.

Parameter	Value
Harmony memory size (HMS)	50
Harmony memory consideration rate (HMCR)	0.7
Pitch adjustment rate (PAR)	0.3
Number of improvisations (NI)	50

The HSA parameters used in this study after the preliminary experiments are shown in Table 3.

The details of the preliminary experiments are discussed below.

3.2.1. *Parameter tuning of the harmony search algorithm*

Given the stochastic factor in meta-heuristic approaches, each problem domain requires a cautious setting of algorithm parameter values. The HSA involves several parameters, such as HMS, HMCR, PAR, and NI. Tuning these parameters is a challenging task because it involves controlling the balance between diversification and intensification. In addition, the parameter values determine whether the HSA finds an optimal or near-optimal solution. Thus, the aim of this section is to study the solution of the HSA over generations under different settings for four important parameters and to select the most suitable parameter for the gene selection problem. After the best parameter setting is discovered, the experimental results and discussion are elaborated in the following section.

Three microarray data-sets with different numbers of features are used in this section: a small data-set with 2000 features (colon), a medium data-set with 12,582 features (mixed-lineage leukaemia [MLL]), and a large data-set with 24,481 features (breast). Each data-set is tested using different parameter settings because the characteristics of each data-set are different.

First, a data-set is selected, and one parameter is selected to be tuned. For example, all parameter values are fixed to the following values to determine the HMS value of the colon data-set: HMCR = 0.7, PAR = 0.3, and NI = 100. The HMS values of 2, 10, 50, 100, 150, and 200 are then investigated by solving the data-set 10 times. After solving all problem instances for the data-set, the best HMS setting is selected based on the highest classification accuracy. A similar procedure is followed to determine the other parameter values (HMCR, PAR, and NI).

3.2.2.1. Harmony memory size. In this section, we determine the effect of the initial HMS on the quality of harmonies and select the most suitable HMS to be used in all the experiments.

Table 4 shows the effects of a variety of HMS values; the value in bold shows the highest classification accuracy obtained for each data-set. Based on Table 4 and the plotted results in Figure 10, we can conclude that decreasing the HMS results in low classification accuracy. This result is attributable to the small population diversity that possibly causes the solution to be trapped in the local optimum. By contrast, the results show that increasing the HMS increases the time required for each iteration and may deteriorate the quality of the solution. The reason for such result is that, during each iteration, the algorithm selects randomly from the HM to construct a new vector. If the HMS increases, the chances of choosing a superior solution from the HM decreases because the HM is populated with many but inferior solutions; this condition may affect the convergence of the HSA.

Table 4. Results of using HSA with different HMS parameter values.

HMS	Colon	MLL	Breast
HMS = 2	56.36	82.65	55.17
HMS = 10	64.6	85.64	54.02
HMS = 50	**69.8**	**86.28**	**57.47**
HMS = 100	63.63	84.69	55.17
HMS = 150	65.45	83.26	54.02
HMS = 200	52.45	79.92	52.87

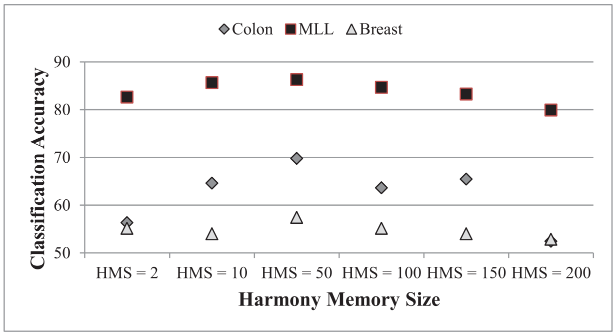


Figure 10. Comparison of different HMS parameter values.

Based on the experimental results presented in Table 4, HMS = 50 is a more suitable parameter value than the other HMS values (2, 10, 50, 100, 150, and 100). Therefore, HMS = 50 is used in all tested instances.

3.2.2.2. Harmony memory consideration rate and pitch adjustment rate. After setting up HMS = 50 (Table 4), we turn our attention to the other parameters, namely, HMCR and PAR. The HMCR variable determines whether the value for the current decision variable in the new vector should come from the HM or be randomly generated. Variability is thus introduced such that the optimisation is not trapped in a local optimum. The PAR variable is similar to the GA mutation rate and is used to change specific elements in the selected individual using PAR probability.

Table 5. Results of using HSA with different HMCR and PAR parameter values.

HMCR	PAR	Colon	MLL	Breast
0.7	0.1	61.03	82.62	56.32
	0.2	63.10	81.52	**57.58**
	0.3	**69.80**	**86.28**	57.47
0.89	0.1	63.63	81.28	56.32
	0.2	65.45	81.25	57.47
	0.3	67.27	81.30	57.47
0.99	0.1	57.23	78.26	51.72
	0.2	57.26	77.62	52.87
	0.3	59.26	80.25	54.02

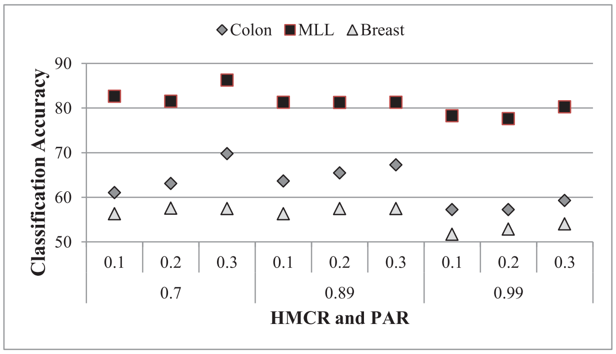


Figure 11. Results of using different parameter values of HMCR and PAR.

We examine several HMCR and PAR values (Table 5). We fix the NI to 50 iterations and HMS to 50. According to the literature, the recommended value for the HMCR is between 0.70 and 0.99, whereas that for the PAR is between 0.1 and 0.3 (Lee & Geem, 2004; Ravikumar Pandi & Panigrahi, 2011). Table 5 shows the experimental results of using different values for the HMCR and PAR parameters. Figure 11 compares different values for the HMCR and PAR parameters that are tested on the three selected data-sets.

Table 5 shows that the best results are obtained when HMCR = 0.7 and PAR = 0.3, that is, on colon and MLL. Competitive results are obtained when HMCR = 0.7 and PAR = 0.2 and 0.3. Therefore, we set the HMCR to 0.7 and the PAR to 0.3 for all the tested instances. Based on the results in Table 5, we can infer the effects of the variation in the HMCR, that is, a large HMCR results in less exploration, and the algorithm further relies on stored values in the HM. In such a case, the algorithm may be trapped in a local optimum. Furthermore, selecting an extremely small HMCR value decreases the efficiency of the HSA, which behaves like a pure random search with less assistance from historical memory. Table 5 also shows the solution quality under different PAR values. We can thus conclude that large PAR values usually improve the best solutions.

Table 6. Classification accuracy values of the improvised solution with various NI values.

No. of iterations	Colon	MLL	Breast
After 10	63.63	67.69	54.02
After 20	65.45	75.38	55.17
After 30	67.27	78.46	56.32
After 40	69.09	81.53	56.32
After 50	69.09	83.07	56.32
After 60	69.09	84.37	56.32
After 70	69.09	84.37	56.32
After 80	69.09	84.95	56.32
After 90	69.09	84.95	56.32
After 100	69.09	84.95	56.32

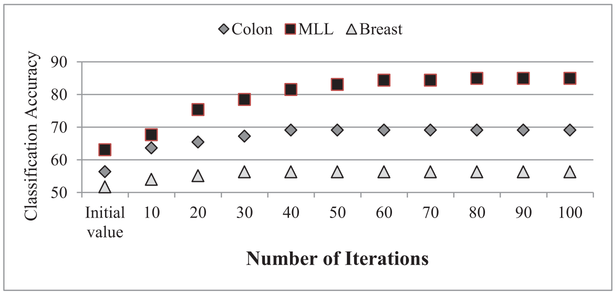


Figure 12. Status of improvised solutions during the run.

3.2.2.3. Number of iterations. In this section, we examine the effect of the NI parameter values on the quality of harmonies. To determine a suitable parameter value for NI, we perform an experiment using various NI values and with fixed values of HMS = 50, HMCR = 0.7, and PAR = 0.3. These experiments help us investigate the behaviour of the HM and decide the suitable maximum number of NI.

The obtained results are presented in Table 6 and plotted in Figure 12. A significant increase in the quality of the solution can be observed during the first 50 iterations. Afterwards (between 50 and 100 iterations), the change in the classification accuracy is small, or almost no improvement can be noted. Thus, NI = 50 is used in this study.

3.3. Comparison of SU-HSA with an HSA using IB1 and NB classifiers

To evaluate the merits of incorporating SU with HSA, we first compare the SU-HSA results with those obtained using HSA without a filter. This comparison allows us to highlight the importance of combining an SU filter with an HSA in a single process. Two classifiers are used (i.e. IB1 and NB) to avoid the examination bias of the classification methods. We employ 10-fold CV (as recommended by Ambroise & McLachlan, 2002) in each data-set, that is, a gene subset is selected using 90% of the instances (treated as a training set). The accuracy of this subset is estimated over the unseen 10% of the data (treated as a test set).

Table 7. Performance of gene selection using a microarray data-set with the IB1 and NB classifiers.

		IB1		NB	
Data-sets		HSA	SU-HSA	HSA	SU-HSA
ALL–AML	#G	**13.96**	24.13	14.33	26.4
	ACC	87.7	99.53	91.21	**100**
ALL–AML-3C	#G	24.7	**22.26**	35.26	24.73
	ACC	77.47	98.92	83.44	**100**
ALL–AML-4C	#G	27.9	**21.4**	33.63	21.73
	ACC	72.94	**97.43**	82.4	97.22
Colon	#G	25.43	22.26	15.73	**9**
	ACC	80.28	87.15	69.8	**87.53**
CNS	#G	12.2	31.63	**8.83**	17.83
	ACC	70.64	**84.44**	72.16	81.42
Lymphoma	#G	20.83	11.8	20.86	**9.9**
	ACC	98.6	**100**	97.03	**100**
MLL	#G	24.76	29.53	32.8	**9.93**
	ACC	84.72	98.73	86.28	**98.97**
Breast	#G	23.56	24.93	**12.43**	14.8
	ACC	68.73	**83.39**	57.58	75.97
Ovarian	#G	31.6	14.93	23.23	**12.3**
	ACC	94.89	**99.94**	92.9	99.65
SRBCT	#G	35.9	**23.23**	26.36	37.53
	ACC	84.16	**100**	86.62	99.89

Note: ACC, average classification accuracy rate (%); |#G|, average number of genes

Table 7 compares SU-HSA and HSA on the 10 data-sets. Each cell in Table 7 shows two pieces of information: the average number of selected genes and the average accuracy over 10 independent runs. The best results in each row are shown in bold. Table 7 shows that the results of the SU-HSA are clearly better than those of the HSA alone for all data-sets in terms of the accuracy. The SU-HSA achieves a lower number of genes in 6 out of 10 instances on both classifiers compared with the HSA. From these results, we infer that the redundant and irrelevant genes are removed efficiently by the SU-HSA.

All the solutions provided by the SU-HSA using IB1 and NB present a classification rate higher than 80% except for the breast data-set (75.97% using the NB classifier). In terms of the number of selected genes, the SU-HSA can obtain less than 30 selected genes on average for nine data-sets, except for the central nervous system (CNS) (31.63) and small, round blue cell tumors (SRBCT) (37.53) data-sets using IB1 and NB, respectively. For the HSA with IB1 and NB, eight and seven data-sets, respectively, have a minimum number of selected genes of less than 30 on average. We believe that the significant superiority of the SU-HSA over the HSA in finding small subsets of genes with high classification accuracy results from the use of SU in selecting the highest effective gene (according to the SU evaluation) when initialising the HM. This scenario is also consistent with the literature on combining the filter with the wrapper in a single approach, which usually achieves better results than using the wrapper alone.

We further validate whether a significant difference exists between the SU-HSA and HSA by conducting a statistical analysis in the form of a Wilcoxon rank test with a 95% confidence level. The *p*-values obtained for both the classification accuracy and the number of genes are shown in Table 8.

Table 8 shows the following:

- The SU-HSA is significantly better than the HSA for all the tested data-sets in terms of classification accuracy (using both classifiers).
- The SU-HSA is significantly better than the HSA for six and eight data-sets involving the IB1 and NB

Table 8. The statistical values (*p*-value) of the Wilcoxon test for the SU-HSA versus the HSA using the IB1 and NB classifiers.

	Accuracy		Number of genes	
Data-sets	SU-HSA versus HSA using IB1	SU-HSA versus HSA using NB	SU-HSA versus HSA using IB1	SU-HSA versus HSA using NB
ALL–AML	0.000	0.000	0.016	0.05
ALL–AML-3C	0.000	0.000	0.277	0.000
ALL–AML-4C	0.000	0.000	0.021	0.000
Colon	0.000	0.000	0.336	0.000
CNS	0.000	0.000	0.000	0.000
Lymphoma	0.000	0.000	0.000	0.000
MLL	0.000	0.000	0.323	0.000
Breast	0.000	0.000	0.931	0.098
Ovarian	0.000	0.000	0.000	0.000
SRBCT	0.000	0.000	0.002	0.000

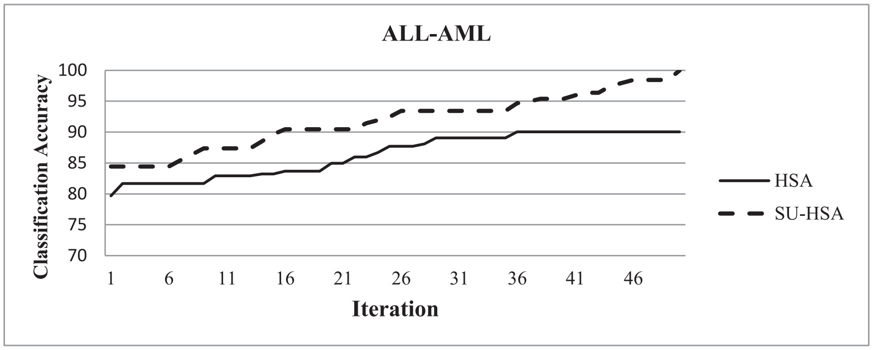


Figure 13. Convergence behaviour of the algorithms on the ALL–AML data-set.

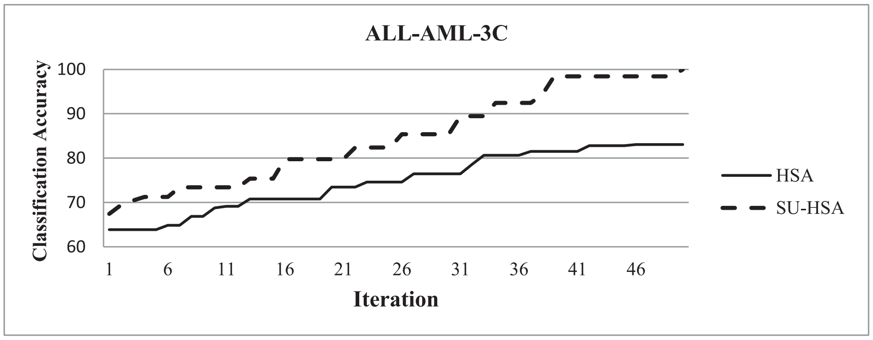


Figure 14. Convergence behaviour of the algorithms on the ALL–AML-3C data-set.

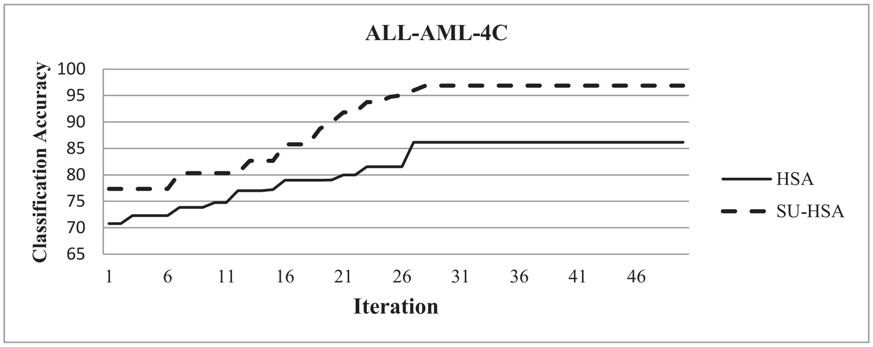


Figure 15. Convergence behaviour of the algorithms on the ALL–AML-4C data-set.

classifiers, respectively, in terms of the number of genes.

Figures 13–22 show the graphs for the HSA and SU-HSA for 50 iterations using the 10 microarray data-sets. These figures represent the number of improvisations versus classification accuracy. The continuous lines represent the HSA, and the dotted lines represent the SU-HSA. The charts reveal that the SU-HSA converges quickly and obtains higher classification accuracy than the HSA for all the data-sets. The main reason why the SU-HSA outperforms the HSA is the two-stage gene selection process for the gene expression data-sets. As the first stage identifies the relevant genes and only selects the genes with the highest SU for use in the next selection stage (the wrapper stage), the genes with low SU values are removed. In the second stage, a wrapper approach (HSA) is implemented to accomplish the gene subset selection process.

3.4. *Comparison with other studies*

The results of the comparison of the proposed approaches with the results of the best-known approaches in the literature for the gene selection problem in microarray data-sets are shown in Table 9. The comparison and subsequent discussion focus on the two key areas addressed in this

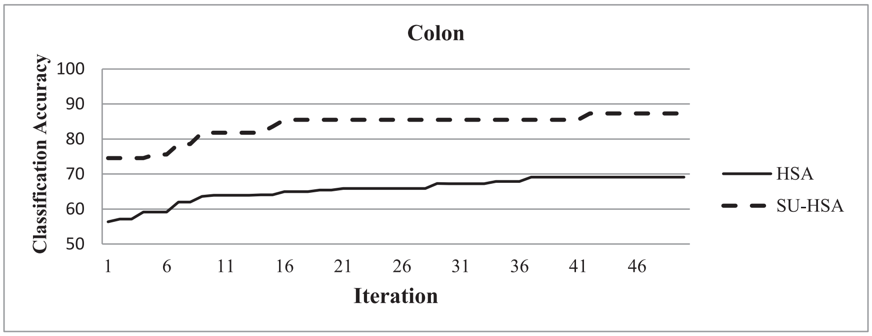


Figure 16 . Convergence behaviour of the algorithms on the colon data-set.

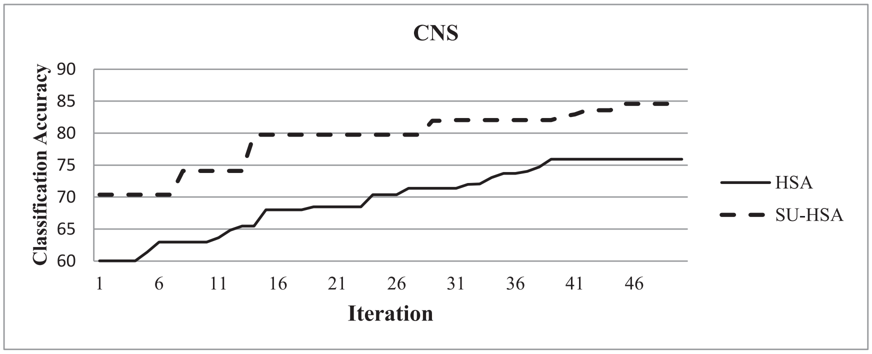


Figure 17. Convergence behaviour of the algorithms on the CNS data-set.

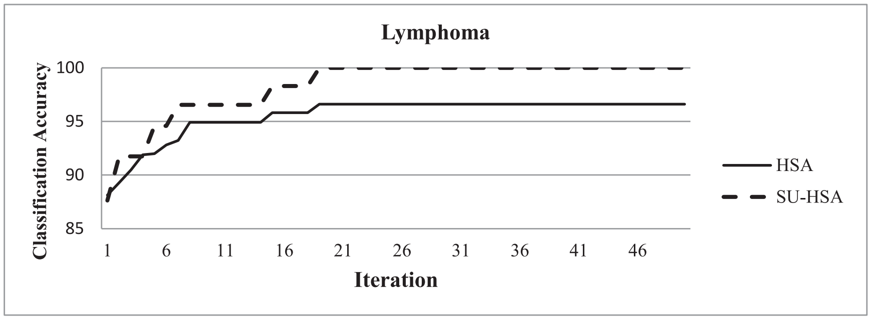


Figure 18. Convergence behaviour of the algorithms on the lymphoma data-set.

research: the classification accuracy and minimal selected genes.

The approaches in the literature that are compared with the methods proposed in this thesis are as follows:

- MBEGA: Markov Blanket-embedded GA for gene selection (Zhu et al., 2007a)
- MRMR-GA: Minimum redundancy–maximum relevancy with a GA (El Akadi et al., 2011)
- MA-C: Correlation-based memetic framework (Senthamarai Kannan & Ramaraj, 2010)
- BPSO-CGA: Binary PSO and a combat GA (Chuang et al., 2011)
- GPSO: Geometric PSO (Talbi et al., 2008)
- BIRSW: Best incremental ranked subset (Ruiz et al., 2006)

Note that not all the algorithms in the comparison have been tested on all the considered data-sets. (We believe this scenario occurs because of the complexity of these instances). We are specifically interested in comparing the SU-HSA with the MBEGA and MA-C because the two latter methods are population based and have been tested on 10 and 8 out of the 10 data-sets, respectively; the other algorithms have been tested on only 3–4 data-sets.

Table 9 clearly shows that in terms of the number of generated genes, the SU-HSA (with IB1 and NB classifiers) can obtain better results on six and eight data-sets compared with the MBEGA and MA-C algorithms, respectively

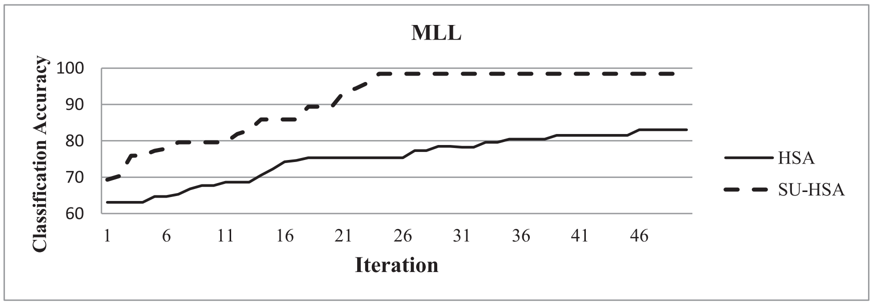


Figure 19. Convergence behaviour of the algorithms on the MLL data-set.

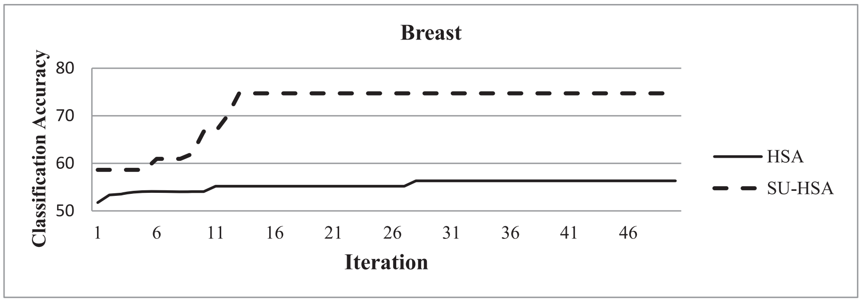


Figure 20. Convergence behaviour of the algorithms on the breast data-set.

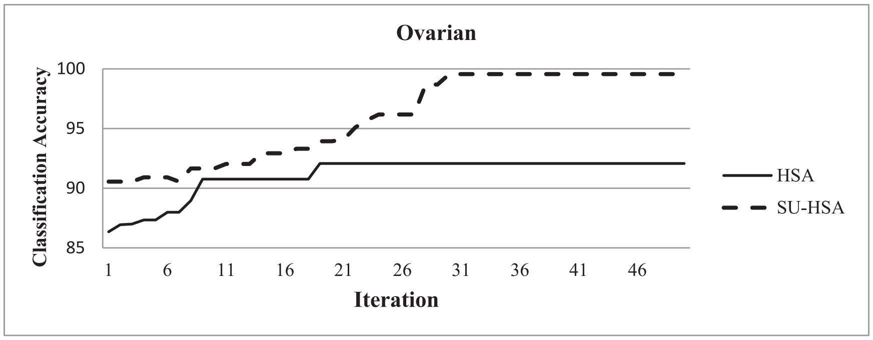


Figure 21. Convergence behaviour of the algorithms on the ovarian data-set.

(with a percentage improvement ranging between 13.02% and 61.70%). In terms of classification accuracy, the SU-HSA outperforms the MBEGA on all of the tested data-sets, with a percentage improvement ranging between 0.23% and 16.93%. Compared with the MA-C, the SU-HSA obtains better classification accuracy on two out of eight data-sets (tied with SRBCT). Although the SU-HSA does not outperform the MA-C on all of the data-sets, we

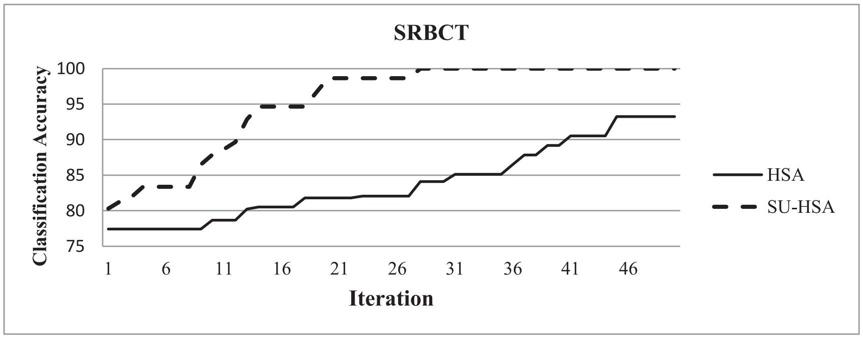


Figure 22. Convergence behaviour of the algorithms on the SRBCT data-set.

Table 9. Results of the comparison of the SU-HSA with state-of-the-art methods.

Data-sets		SU-HSA IB1	SU-HSA NB	MBEGA	MRMR-GA	MA-C	BPSO-CGk	GPSO	BIRSW
ALL–AML	|#G|	24.13	26.4	12.8	15	387	300	3	**2.5**
	ACC	99.53	**100**	95.89	**100**	99.56	**100**	97.38	93.04
ALL–AML-3C	|#G|	22.26	24.73	**18.1**	–	394	–	–	–
	ACC	98.92	**100**	96.64	–	99.53	–	–	–
ALL–AML-4c	|#G|	**21.4**	21.73	26.2	–	386	–	–	–
	ACC	97.43	97.22	91.93	–	**98.61**	–	–	–
Colon	|#G|	22.26	9	24.5	15	–	214	**2**	3.5
	ACC	87.15	87.53	85.66	98.39	–	96.7	**100**	85.48
CNS	|#G|	31.63	**17.83**	20.5	–	374	–	–	–
	ACC	84.44	81.42	72.21	–	**97.78**	–	–	–
Lymphoma	|#G|	11.8	**9.9**	34.3	15	–	196	–	10.3
	ACC	**100**	**100**	97.68	98.96	–	**100**	–	82.14
MLL	|#G|	29.53	**9.93**	32.1	–	108	–	–	–
	ACC	98.73	98.97	94.33	–	**100**	–	–	–
Breast	|#G|	24.93	14.8	14.5	–	183	–	**4**	–
	ACC	83.39	75.97	80.74	–	**95.26**	–	86.35	–
Ovarian	|#G|	14.93	12.3	**9**	–	247	–	4	–
	ACC	99.94	99.65	99.71	–	**100**	–	99.4	–
SRBCT	|#G|	**23.23**	37.53	60.7	–	526	880	–	–
	ACC	**100**	99.89	99.23	–	**100**	**100**	–	–

Note: ACC, average classification accuracy rate (%); |#G|, average number of genes, –, not available

believe that this result is due to the high number of genes generated by the MA-C and used to measure classification accuracy. Theoretically, more genes could provide a higher classification accuracy; however, the higher number of genes generated will not only slow down the learning process, but could also lead to the inclusion of some irrelevant genes that might give incorrect results. Given the high difference between the SU-HSA and MA-C in terms of the generated number of genes and the small difference in terms of classification accuracy, we can conclude that the SU-HSA performs better than the MA-C based on the eight data-sets used in the comparison. Again, we believe that this result is due to the combination of the filter and wrapper approaches.

In terms of the number of generated genes, the SU-HSA outperforms the MRMR-GA, BPSO-CGk, and BIRSW on two, four, and one data-sets, respectively. Note that these methods (MRMR-GA, BPSO-CGk, and BIRSW) have not been tested on all of the considered instances. With respect to classification accuracy, the SU-HSA outperforms the MRMR-GA, GPSO, and BIRSW on one, two, and three instances, respectively. The SU-HSA achieves the same classification accuracy on three out of the four instances compared with the BPSO-CGk. Although the SU-HSA does not outperform GPSO in terms of generated genes, the SU-HSA nevertheless achieves a high classification rate for two instances, and GPSO is only tested on four instances.

In sum, the SU-HSA obtains results that are more competitive (if not better in some instances) than those of the best methods in the literature. The SU-HSA outperforms the best-known results in some instances. These positive results indicate that the SU-HSA efficiently deals with problems involving very large numbers of genes. In our opinion, this result is due to the following reasons. (1) The SU can address different problem instances by removing irrelevant genes. Removing some of the irrelevant genes allows the wrapper phase (HSA) to explore the search space efficiently. (2) As gene selection problems are very difficult to solve and have many local optima, the HSA is more effective in diversifying the search for solutions by exploring different regions and considering all existing solutions in the HM instead of using only two solutions (parents), as in the GA. In gene selection problems, some genes are common in several solutions. By considering all solutions, the HSA can thus combine these common genes with other genes to obtain a high-quality solution. Overall, our approach does not rely on complicated search approaches and is simple to implement regardless of the nature and complexity of the problems.

4. Conclusions

In this study, we propose a method for gene selection by combining an SU filter and an HSA wrapper. This method has two stages. In the first stage, an SU filter selects the gene subsets with the highest SU score to initialise the HM. In the second stage, the HSA wrapper (which combines the HSA search strategy with the classifier [i.e. IB1 and NB]) is used in gene subset selection. The experimental results demonstrate the superiority of the SU-HSA over the HSA alone on almost all the tested data-sets. The contribution of this study is the development of a gene selection method that combines both the filter and the wrapper methods; that is, the SU and HSA are hybridised. The proposed method can select the most appropriate subset of genes and achieve a classification accuracy that compares competitively with other available approaches in the literature.

Funding

This paper is based on the work supported by the Malaysia Ministry of Education research grant, in part by [grant number FRGS/1/2012/SG05/UKM/03/3]; [grant number UKM-TT-07-FRGS0157-2010].

Notes on contributors

Salam Salameh Shreem received his BSc degree in computer information system from Al-Zaytoonah University of Jordan, Amman, Jordan, in 2004, and his MSc degree in computer science from Al-Balqa' Applied University, Al-Salt, Jordan, in 2007. He did his PhD in computer science at Universiti Kebangsaan Malaysia (UKM) in January 2014. His research interests are mainly directed to meta-heuristic optimisation approaches and data mining problems.

He is currently an Assistant Professor with the College of Computing and Informatics, Saudi Electronic University, Riyadh, Saudi Arabia. His research interests are mainly directed to meta-heuristic optimisation approaches and data mining problems.

Salwani Abdullah obtained her BSc degree in computer science from Universiti Teknologi Malaysia and her master's degree specialising in computer science from UKM. She did her PhD degree in computer science at University of Nottingham, United Kingdom. Now, she is a professor at the Faculty of Information Science and Technology, UKM. Her research interest falls under Artificial Intelligence and Operation Research, particularly in meta-heuristic algorithms in optimisation areas that involve different real-world applications and optimisation problems, such as university timetabling, job shop scheduling, nurse rostering, space allocation, and data mining tasks.

Mohd Zakree Ahmad Nazri is a senior lecturer at the Center for Artificial Intelligence Technology, Faculty of Information Science & Technology, UKM. He is a research fellow in the Data Mining and Optimisation research group. He is the head of project for numerous research funded by the government and industry. He received his PhD and master's degrees in computer science from Universiti Teknologi Malaysia. He obtained his bachelor degree in system science and management from UKM. His current research interests include soft computing, decision support, and optimisation. He has contributed articles to *Information Science*, *Computers and Industrial Engineering*, and *Natural Computing*.

References

Agrawal, R., & Bala, R. (2007). A hybrid approach for selection of relevant features for microarray datasets. *International Journal of Computer, Information Science and Engineering, 1*(4), 196–202.

Al-Betar, M.A., & Khader, A.T. (2008). A harmony search algorithm for university course timetabling. *Annals of Operations Research, 194*(1), 1–29.

Al-Betar, M.A., Khader, A.T., & Liao, I.Y. (2010). A harmony search with multi-pitch adjusting rate for the university course timetabling. *Recent Advances in Harmony Search Algorithm, 270*, 147–161.

Alia, O.M., Mandava, R., & Aziz, M.E. (2010). A hybrid harmony search algorithm for MRI brain segmentation. *Evolutionary Intelligence, 4*(1), 1–19.

Alia, O., Mandava, R., Ramachandram, D., & Aziz, M.E. (2009a). Harmony search-based cluster initialization for fuzzy c-means segmentation of MR images. In *TENCON 2009–2009 IEEE region 10 conference* (pp. 1–6). Singapore: IEEE.

Alia, O.M., Mandava, R., Ramachandram, D., & Aziz, M.E. (2009b). Dynamic fuzzy clustering using harmony search with application to image segmentation. In *International Symposium on Signal Processing and Information Technology* (pp. 538–543). Ajman: IEEE.

Amaldi, E., & Kann, V. (1998). On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science, 209*(1–2), 237–260.

Ambroise, C., & McLachlan, G.J. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences, 99*(10), 6562–6566.

Ayvaz, M.T. (2007). Simultaneous determination of aquifer parameters and zone structures with fuzzy c-means clustering and meta-heuristic harmony search algorithm. *Advances in Water Resources, 30*(11), 2326–2338.

Bermejo, P., Gámez, J.A., & Puerta, J.M. (2011). A GRASP algorithm for fast hybrid (filter–wrapper) feature subset selection in high-dimensional datasets. *Pattern Recognition Letters, 32*(5), 701–711.

Cheng, Y., Li, L., Lansivaara, T., Chi, S., & Sun, Y. (2008). An improved harmony search minimization algorithm using different slip surface generation methods for slope stability analysis. *Engineering Optimization, 40*(2), 95–115.

Chuang, L.Y., & Yang, C.H. (2009). Tabu search and binary particle swarm optimization for feature selection using microarray data. *Journal of Computational Biology, 16*(12), 1689–1703.

Chuang, L.Y., Yang, C.H., & Li, J.C. (2011). A hybrid BPSO-CGA approach for gene selection and classification of microarray data. *Journal of Computational Biology, 19*(1), 1–14.

Chuang, L.-Y., Yang, C.-H., Wu, K.-C., & Yang, C.-H. (2011). A hybrid feature selection method for DNA microarray data. *Computers in Biology and Medicine, 41*(4), 228–237.

Chuang, L.-Y., Yang, C.-S., Wu, K.-C., & Yang, C.-H. (2011b). Gene selection and classification using Taguchi chaotic binary particle swarm optimization. *Expert Systems with Applications, 38*(10), 13367–13377.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis, 1*(3), 131–156.

Duval, B., & Hao, J.K. (2010). Advances in metaheuristics for gene selection and classification of microarray data. *Briefings in Bioinformatics, 11*(1), 127–141.

Duval, B., Hao, J.-K., & Hernandez Hernandez, J.C. (2009). A memetic algorithm for gene selection and molecular classification of cancer. *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation* (pp. 201–208). Montreal: ACM.

El-Abd, M. (2012). Performance assessment of foraging algorithms vs. evolutionary algorithms. *Information Sciences, 182*(1), 243–263.

El Akadi, A., Amine, A., El Ouardighi, A., & Aboutajdine, D. (2011). A two-stage gene selection scheme utilizing MRMR filter and GA wrapper. *Knowledge and Information Systems, 26*(3), 487–500.

Fesanghary, M., Damangir, E., & Soleimani, I. (2009). Design optimization of shell and tube heat exchangers using global sensitivity analysis and harmony search algorithm. *Applied Thermal Engineering, 29*(5–6), 1026–1031.

Forsati, R., Mahdavi, M., Shamsfard, M., & Reza Meybodi, M. (2013). Efficient stochastic algorithms for document clustering. *Information Sciences, 220*, 269–291.

Geem, Z.W. (2007). Harmony search algorithm for solving sudoku. In Apolloni, B., Howlett, R. and Jain, L. (Eds.), *Knowledge-Based Intelligent Information and Engineering Systems* (pp. 371–378). Berlin, Heidelberg: Springer.

Geem, Z.W. (2009a). *Harmony search algorithms for structural design optimization* (239th ed.). Berlin: Springer.

Geem, Z.W. (2009b). Particle-swarm harmony search for water network design. *Engineering Optimization,* 41(4), 297–311.

Geem, Z., & Choi, J.Y. (2007). Music composition using harmony search algorithm. *Applications of Evolutionary Computing, 4448*, 593–600.

Geem, Z.W., Kim, J.H., & Loganathan, G. (2001). A new heuristic optimization algorithm: harmony search. *Simulation, 76*(2), 60–68.

Hall, M.A. (1999). *Correlation-based feature selection for machine learning* (PhD thesis). University of Waikato, Hamilton, New Zealand.

Hasan, B.H.F., Abu Doush, I., Al Maghayreh, E., Alkhateeb, F., & Hamdan, M. (2014). Hybridizing harmony search algorithm with different mutation operators for continuous problems. *Applied Mathematics and Computation, 232*, 1166–1182.

Huang, C.L. (2009). ACO-based hybrid classification system with feature subset selection and model parameters optimization. *Neurocomputing, 73*(1), 438–448.

Kohavi, R., & John, G.H. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1–2), 273–324.

Lee, K.S., & Geem, Z.W. (2004). A new structural optimization method based on the harmony search algorithm. *Computers & Structures, 82*(9–10), 781–798.

Lee, K.S., & Mun, S. (2014). Improving a model for the dynamic modulus of asphalt using the modified harmony search algorithm. *Expert Systems with Applications, 41*(8), 3856–3860

Li, T., Zhang, C., & Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics, 20*(15), 2429–2437.

Lin, S.W., & Chen, S.C. (2012). Parameter determination and feature selection for C4.5 algorithm using scatter search approach. *Soft Computing – A Fusion of Foundations, Methodologies and Applications, 16*(1), 63–75.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining* (454th ed.). Boston, MA: Kluwer Academic.

Lozano, M., & García-Martínez, C. (2010). Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research, 37*(3), 481–497.

Mahdavi, M., Chehreghani, M.H., Abolhassani, H., & Forsati, R. (2008). Novel meta-heuristic algorithms for clustering web documents. *Applied Mathematics and Computation, 201*(1–2), 441–451.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation, 188*(2), 1567–1579.

Maheri, M.R., & Narimani, M. (2014). An enhanced harmony search algorithm for optimum design of side sway steel frames. *Computers & Structures, 136*, 78–89.

Nekooei, K., Farsangi, M.M., Nezamabadi-Pour, H., & Lee, K.Y. (2013). An improved multi-objective harmony search for optimal placement of DGs in distribution systems. *IEEE Transactions on Smart Grid, 4*(1), 557–567.

Panchal, A. (2009). Harmony search in therapeutic medical physics. In Z.W. Geem (Ed.), *Music-inspired harmony search algorithm* (pp. 189–203). Berlin: Springer.

Ravikumar Pandi, V., & Panigrahi, B.K. (2011). Dynamic economic load dispatch using hybrid swarm intelligence based harmony search algorithm. *Expert Systems with Applications, 38*(7), 8509–8514.

Ruiz, R., Riquelme, J.C., & Aguilar-Ruiz, J.S. (2006). Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition, 39*(12), 2383–2392.

Senthamarai Kannan, S., & Ramaraj, N. (2010). A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm. *Knowledge-Based Systems, 23*(6), 580–585.

Sirjani, R., Mohamed, A., & Shareef, H. 2012. Optimal allocation of shunt Var compensators in power systems using a novel global harmony search algorithm. *International Journal of Electrical Power & Energy Systems, 43*(1): 562–572

Sun, X., Liu, Y., Xu, M., Chen, H., Han, J., & Wang, K. (2012). Feature selection using dynamic weights for classification. *Knowledge-Based Systems, 37*, 541–549.

Talbi, E.G., Jourdan, L., Garcia-Nieto, J., & Alba, E. (2008). Comparison of population based metaheuristics for feature selection: Application to microarray data classification. In *International Conference on Computer Systems and Applications* (pp 45–52). Doha: IEEE.

Tamer Ayvaz, M. (2009). Application of harmony search algorithm to the solution of groundwater management models. *Advances in Water Resources, 32*(6), 916–924.

Vasebi, A., Fesanghary, M., & Bathaee, S. (2007). Combined heat and power economic dispatch by harmony search algorithm.

International Journal of Electrical Power & Energy Systems, 29(10), 713–719.

Wang, L., Yang, R., Xu, Y., Niu, Q., Pardalos, P.M., & Fei, M. (2013). An improved adaptive binary Harmony Search algorithm. *Information Sciences, 232*, 58–87.

Yildiz, A.R. (2012). A comparative study of population-based optimization algorithms for turning operations. *Information Sciences, 210*, 81–88.

Yusta, S.C. (2009). Different metaheuristic strategies to solve the feature selection problem. *Pattern Recognition Letters, 30*(5), 525–534.

Zhang, H., & Sun, G. (2002). Feature selection using tabu search method. *Pattern Recognition, 35*(3), 701–711.

Zhu, Z., Ong, Y.-S., & Dash, M. (2007a). Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition, 40*(11), 3236–3248.