Chaotic Harmony Search based Multi-objective Feature Selection for Classification of Gene Expression Profiles

1st Aiguo Wang
School of Electronic Information Engineering
Foshan University
Foshan, China
wangaiguo2546@163.com

2nd Huancheng Liu
School of Electronic Information Engineering
Foshan University
Foshan, China
liuhuancheng83@163.com

3rd Guilin Chen
School of Computer and Information Engineering
Chuzhou University
Chuzhou, China
glchen@chzu.edu.cn

2021 IEEE 9th International Conference on Bioinformatics and Computational Biology (ICBCB) | 978-1-6654-1500-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICBCB52223.2021.9459222

***Abstract*—How to effectively select a subset of discriminant features from the high-dimensional low- sample-size microarray gene expression profiles remains crucial and meaningful for the bioinformatics analysis tasks such as locating disease genes and building classifiers for cancer diagnosis. Though meta-heuristic harmony search algorithm has been used for feature selection, it suffers from entrapment in local optima and low convergence speed. To this end, we propose a hybrid chaotic harmony search based multi-objective feature selection method, which uses the chaotic map to replace the parameter of harmony search during the optimization process. Specifically, the minimum redundancy maximum relevancy feature selector is first used to pre-select a subset of relevant features. Then, the chaotic harmony search is employed on the reduced feature set to find an optimal feature subset, where the fitness of a candidate solution is evaluated by a multi-objective formulation. Finally, extensive comparative experiments against its competitors, including six filter and four wrapper feature selection methods, are conducted on six public microarray datasets. Results show that the proposed method obtains higher classification accuracy. Besides, the convergence analysis indicates its efficiency.**

Keywords—gene expression profiles, feature selection, chaotic optimization, harmony search

I. INTRODUCTION

Microarray technology enables us to obtain the expression profiles of thousands of genes simultaneously, which provides an objective way to locate disease genes and predict cancers at the molecular level [1]. Accordingly, researchers have used various statistical analysis and machine learning models to analyze the gene expression profiles. However, microarray data are typically characterized by small sample size (tens of samples) and high-dimensionality (thousands of genes), which suffers a great deal from the curse of dimensionality and limits the statistical power for practical use. One feasible way to alleviate this problem is to reduce the dimensionality with dimensionality reduction methods [2, 3].

Dimensionality reduction methods can be broadly grouped into feature extraction and feature selection methods, where the former extract a group of new features (a combination of the original features) and the latter select a subset of features [3, 4]. Obviously, feature selection remains a priority for the subsequent tasks such as knowledge discovery, identifying biomarkers, and interpretability. Feature selection, called gene selection in the context of microarray data, aims to select a small subset of features from the original feature space by removing irrelevant and redundant features while maintaining the classification accuracy [4]. Accordingly, there are a wealth of feature selection methods available for use. According to the results returned by a feature selector, existing feature selection methods can be grouped into feature ranking and feature subset methods, where the latter return a final feature subset, while the former return a ranked list of the original features and an extra step is needed to determine how many features to select. According to whether a classification model is involved in the feature selection process, we can group them into filter, wrapper, embedded, and hybrid methods [5, 6]. Filter methods are independent of a classification model and use the designed metrics to measure the quality of a feature or a feature subset. Commonly used metrics include consistency-, distance-, dependency-, and information theoretic criterion-based metrics [4]. Filter methods are flexible in combination with different classification models and generally have lower computational complexity. In contrast, wrapper methods use a classification model to evaluate the goodness of a candidate feature [6]. They usually take as the evaluation metrics the classification accuracy, error rates, and area under the curve. Benefiting from its interaction with the classification model, wrapper methods generally provide better accuracy, but with higher computational cost. Embedded methods are a special case of wrapper methods and they obtain the feature subset during building classifiers. Hybrid methods are a combination of filter and wrapper methods in order to take advantage of both methods. One commonly used scheme is to first use a filter to remove irrelevant features and then use a wrapper to obtain the final feature subset [7].

Particularly, in developing a feature selection method, the search strategy largely determines its performance. Since the sequential search methods (e.g., sequential forward/backward

This work was supported in part by the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2020A1515011499), the Major Special Projects of Anhui Province (Grant No. 201903A06020026), and the Anhui Provincial Natural Science Foundation (Grant No. 1908085MF211).

selection, and sequential floating selection) easily suffer from entrapment in local optima, researchers have explored meta-heuristic algorithms having the global-search capability for feature selection, among which harmony search (HS) has been applied due to its simplicity and flexibility [8, 9]. However, as most meta-heuristic algorithms, HS is also easily trapped into local optima. To this end, we herein present a hybrid chaotic harmony search based feature selection method that uses the chaotic map to control the optimization process [10]. Specifically, we first use the minimum redundancy maximum relevancy (mRMR) feature selector to pre-select a subset of features, and then use the chaotic HS with a multi-objective fitness function to obtain the final features from the results of mRMR. The main contributions of this study include the following. (1) We present a hybrid feature selection method and incorporate the chaos theory into harmony search. Due to the quasi-stochastic property, ergodicity, and sensitivity against initial conditions of the chaotic map, this helps to improve the performance of harmony search algorithm. (2) We present a multi-objective fitness function to balance the predictive power and the size of a candidate feature subset. (3) We conduct experiments and compare the proposed method with other ten feature selectors. Results demonstrate the superiority of the proposed method over its competitors.

II. The Proposed Method

A. Harmony Search based Feature Selection

Harmony search, one of the meta-heuristic algorithms, is inspired by the improvisation process of a group of music players. It has the advantage of flexibility, simplicity, and a fast convergence rate [9], and has a wide range of applications such as optimization, machine learning, and complex system control. Harmony search algorithm mainly consists of two phases: initialization phase and iteration phase, where the former randomly generates a pool of initial solutions and the latter improves the candidate solutions. For HS-based feature selection, it works with the following steps. Algorithm 1 also presents the pseudo-code, where rand( ) generates a random number between 0 and 1 and randInt(a, b) generates an integer between a and b.

(1) Initializing harmony memory. Randomly generate a pool of Harmony Memory Size (*HMS*) harmonies within the feasible solution space, and store them in a matrix *HM*, called harmony memory. Each column is a feature (musician), and each row is a solution (harmony) and has a dimension N equal to the number of features. For feature selection, each row is a N-dimensional vector that codes a candidate feature subset, and its elements are configured with binary values with one corresponding to the selection of a feature.

(2) Improvising a Harmony. Improvising a harmony H_{new} with *memory consideration*, *pitch adjustment*, and *random consideration*. Specifically, for each element of H_{new}, if the random number between 0 and 1 is greater than the harmony memory consideration rate (*HMCR*), the pitch is determined by the random consideration operation (line 9); otherwise, a pitch is picked from the memory (line5), and if a random number is less than the pitch adjusting rate (*PAR*), the pitch adjustment operation sets the binary value to its opposite.

Algorithm 1. HS-based Feature Selection

Input: Training set *Data*, *HMS*, *HMCR*, *PAR*, *NI*;
Output: Final selected features *S*

```
1   Randomly generate HMS solutions and store them in HM;
2   for count = 1 to NI do //number of improvisation
3     H_new = zeros(1, N); //initialize a new harmony
4     for t = 1 to N do //for each pitch (feature)
5       if rand( ) ≤ HMCR do
6         H_new(1, t) = HM(randInt(1, HMS), t);
7         if rand( ) < PAR do //pitch adjustment
8           H_new(1, t) = 1 - H_new(1, t);
9       else
10        if rand( ) < 0.5 do
11          H_new(1, t) = 1;
12    idx = min(fitness(HM)); //worst harmony index
13    if fitness(H_new) > fitness(HM(idx, :)) do
14      HM(idx, :) = H_new;
15  final = max(fitness(HM)); //best harmony index
16  obtain S with the final-th harmony;
17  Return S;
```

(3) Updating harmony memory. If the fitness of H_{new} is higher than the worst fitness of harmony H_{idx} in *HM*, then H_{idx} is replaced by H_{new}. Since feature selection aims to minimize the data dimensionality while maintaining the classification accuracy, we take it as a multi-objective problem and use the following function to measure the fitness of a harmony H in order to achieve the balance between the two objectives.

$$fitness(H) = \alpha * acc + (1-\alpha) * (1 - \frac{|S|}{N}) \quad (1)$$

, where S is the feature subset coded by H, *acc* is the accuracy in percent of a classifier trained on S, and α is the tradeoff between the two components.

(4) Stopping criteria. The improvisation process ends when a pre-defined maximum number of improvisations (*NI*) is reached.

B. Chaotic Harmony Search based Feature Selection

Predefined fixed parameters are typically adopted in the classical harmony search algorithm, and it is a non-trivial task to determine a good parameter value. Accordingly, there are studies that use the dynamic parameter adjustment schemes. Motivated by the improvisation process, researchers proposed to increase the harmony memory and *HMCR*, and decrease the *PAR* with the iteration t ($1 \le t \le NI$) at run time, where linear scheme is commonly used, as shown in Eqs. (2)-(4),

$$HMS_t = HMS_{\min} + \frac{t}{NI}(HMS_{\max} - HMS_{\min}) \quad (2)$$

$$HMCR_t = HMCR_{\min} + \frac{t}{NI}(HMCR_{\max} - HMCR_{\min}) \quad (3)$$

$$PAR_t = PAR_{\max} - \frac{t}{NI}(PAR_{\max} - PAR_{\min}) \quad (4)$$

Although the above scheme generally achieves improved performance, it is still not easy to set their values and HS may get stuck in local optima. In order to balance the exploitation and exploration, the chaotic optimization can be used to boost

the performance. Due to the ergodicity and non-repetition of chaos, chaotic algorithm can search the solution space more efficiently than the probability-dependent stochastic methods. The key idea of chaotic harmony search is to replace the HS parameters with the chaotic pseudorandom sequences. Herein, we propose to tune *HMCR* and *PAR* with the logistic map (Eq. 5) to obtain the chaotic sequence,

$$C_{t+1} = 4 * C_t * (1 - C_t),\ t \in \{0,1,2,...,N\} \qquad (5)$$

, where C_0 is a randomly generated initial value. Algorithm 2 presents the pseudo-code, where the chaotic sequences of *HMCR* and *PAR* are *CH* and *CP*, respectively (see line 5). Besides, how to add the dynamic harmony memory operation in the harmony search process is given in line 16.

C. Proposed Feature Selection Method

The proposed method, named mRMR-CHS, is a hybrid combination of filter and wrapper methods and consists of two stages to obtain the final features. First, the top ranked *Q* features are selected from the original feature space with mRMR. This helps greatly reduce the search space for HS. Afterwards, chaotic harmony search based feature selection is adopted on the reduced feature set to obtain the final features. Fig. 1 presents the proposed feature selection method and how to use it in the classification framework.

Algorithm 2. Chaotic HS-based Feature Selection

Input: Training set *Data*, *HMS*, *HMCR*, *PAR*, *NI*;

Output: Final selected features *S*

```
1   Randomly generate HMS solutions and store them in HM;
2   for count = 1 to NI do //number of improvisation
3       H_new = zeros(1, N); //initialize a new harmony
4       for t = 1 to N do //for each pitch (feature)
5           get CH_t and CP_t using the chaotic map
6           if rand( ) ≤ CH_t do
7               H_new(1, t) = HM(randInt(1, HMS), t);
8               if rand( ) < CP_t do //pitch adjustment
9                   H_new(1, t) = 1 - H_new(1, t);
10          else
11              if rand( ) < 0.5 do
12                  H_new(1, t) = 1;
13      idx = min(fitness(HM)); //worst harmony index
14      if fitness(H_new) > fitness(HM(idx, :)) do
15          HM(idx, :) = H_new;
16      generate (HMS_t-HMS_min) new harmonies for HM;
17  final = max(fitness(HM)); //best harmony index
18  obtain S with the final-th harmony;
19  Return S;
```

III. Experimental Results and Analysis

A. Experimental Dataset

To evaluate the proposed method, extensive comparative experiments are conducted on six public microarray datasets that cover both binary and multi-classes cases [6]. Table I gives their characteristics. The last column "#SGR" denotes the ratio of the number of samples to the number of genes.

COLON: It consists of 62 samples and has 2000 genes. The task is to distinguish between cancer and normal samples.

DLBCL: It is about the diffuse large-B-cell lymphoma data and has 77 samples. Each sample contains 7129 genes.

ALLAML: It has 72 samples from the acute lymphoma leukemia (ALL) and acute myeloid leukemia (AML) tissues. Each sample has 7129 genes. The task is to distinguish the two types of leukemia.

PROSTATE: It has 50 normal and 52 prostate samples. The number of genes is 12600. The task is to distinguish the tumor from normal samples.

LEUKEMIA: It has 5327 genes and 72 samples. The task is to classify three subtypes (i.e., B-cell ALL, T-cell ALL, and AML) of leukemia.

SRBCT: It includes 83 samples about four different types of childhood tumors. Each sample consists of 2308 genes.

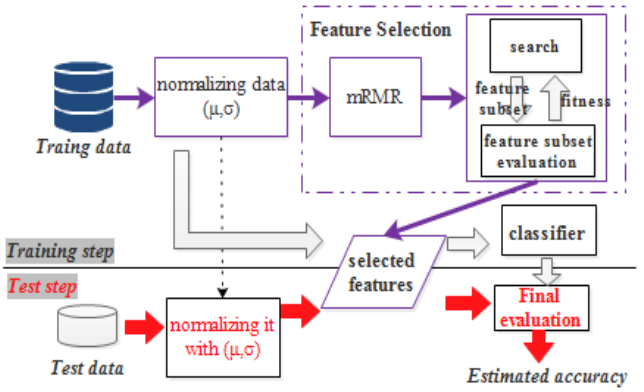

Fig. 1. Flowchart of the feature selection and classification.

TABLE I. Experimental Datasets

Dataset	#Classes	#Samples	#Genes	#SGR
COLON	2	62 (40/22)	2000	0.031
DLBCL	2	77 (58/19)	7129	0.011
ALLAML	2	72 (25/47)	7129	0.010
PROSTATE	2	102 (52/50)	12600	0.008
LEUKEMIA	3	72 (38/9/25)	5327	0.014
SRBCT	4	83 (29/25/11/18)	2308	0.036

B. Experimental Setup

To generate independent training sets and test sets and to avoid selection bias in gene extraction, nested cross validation is adopted [11]. Specifically, external ten-fold cross validation is used, where each one of the ten folds is retained as a test set to evaluate the quality of the final selected features and the remaining folds are used as the training set to select features and to train a classifier. The final result is the average of the ten results. Particularly, the selection of genes and the training of a classifier are only performed on the training set. Each sample of the training set is transformed to zero mean and unit standard deviation, and we also normalize the test set with the means and standard deviations of the training set. Fig. 1 gives the overall framework for gene selection and microarray data classification. The lower part is the test step and the upper part denotes the training step. In feature selection, to determine the fitness of a candidate feature subset with Eq. (1), the inner leave one out cross validation on the training set is used. We here use *k*-nearest-neighbor (KNN) to calculate the fitness of a feature subset and to evaluate the quality of the final selected

features [12, 13]. In the experiments, *Accuracy* (*Acc*) and *F*1 are used as the performance metrics. Given $C = \{C_1, C_2, \ldots, C_{|C|}\}$ to denote a label set with $|C|$ different classes, *F*1 is the harmonic mean of *precision* and *recall*.

$$F1 = \frac{2 * precision * recall}{precision + recall} \tag{6}$$

Precision is the average of the correctly classified sample for each class.

$$Precision = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{T_i}{NP_i} \tag{7}$$

, where T_i is the number of samples from class C_i that are correctly classified, and NP_i is the number of samples predicted with class C_i.

Recall is the percentage of correctly retrieved samples for each class.

$$Recall = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{T_i}{NT_i} \tag{8}$$

, where NT_i equals the number of samples from class C_i.

$$Accuracy = \sum_{i=1}^{|C|} T_i \Big/ \sum_{i=1}^{|C|} NT_i \tag{9}$$

As for the feature selectors, we include the commonly used methods ReliefF, mutual information maximization (MIM), min-redundancy max-relevance (mRMR), conditional mutual information maximization (CMIM), joint mutual information (JMI), as well as fast correlation based filter (FCBF) for comparisons [14]. Since ReliefF, MIM, CMIM, MRMR, and JMI return a ranked list of features, we experimentally choose the twenty-five top-ranked genes to get the final feature subset [15, 16]. Besides, we compare mRMR-CHS with other four harmony search-based feature selection methods, called regHS, dyHS, mRMR-regHS, and mRMR-dyHS. Specifically, regHS and dyHS select features from the original feature space, while mRMR-regHS and mRMR-dyHS first use mRMR to filter out irrelevant features and then select features within the reduced feature space. Besides, regHS and mRMR-regHS use the fixed predefined parameter values, and dyHS and mRMR-dyHS use a dynamic parameter adjustment scheme. In the experiments, for mRMR-regHS, mRMR-dyHS, and mRMR-CHS, we first use mRMR to pre-select one-hundred fifty features and then use harmony search to obtain the final features. Table II presents the parameter settings of HS used in the experiments.

TABLE II. PARAMETER SETTINGS

Algorithm	Parameter	Value	Parameter	Value
regHS and *mRMR-regHS*	*Memory size* *HMCR* *PCR*	30 0.8 0.3	*Max iterations NI* *Fitness weight α*	100 0.9
dyHS and *mRMR-dyHS*	*Memory size* *HMCR* *PCR*	30-50 0.5-0.9 0.3-0.1		
mRMR-CHS	*Memory size*	30-50		

C. Classification Performance

Table III presents the classification performance of the proposed method and the six compared filter methods. The best accuracy achieved on each microarray data is shown in bold, and the last row "average" gives the averaged results of the datasets. The "w/o" denotes the results that are obtained without using feature selection. We observe that mRMR-CHS obtains the best results on four datasets and achieves comparable performance to the best results on the remaining two datasets. For example, mRMR-CHS obtains 100.00% accuracy on *SRBCT*, which is higher than the 91.57% of reliefF, 98.80% of MIM, 98.80% of mRMR, 95.18% of CMIM, 98.80% of JMI, and 93.98% of FCBF. Second, we observe that mRMR-CHS achieves comparable performance to or outperforms mRMR on all the datasets. This indicates that the selected features of mRMR still contains irrelevant and redundant features.

We then compare mRMR-CHS with other four harmony search based feature selection methods. Table IV shows the experimental results on the datasets. The last row "average" denotes the averaged results on all the datasets. The columns "#genes" and "#std" denote the mean and standard deviation of the number of selected features of the external ten-fold cross validation, respectively. The best result on each dataset is shown in bold. We observe that mRMR-CHS outperforms its competitors in five out of the six datasets. Second, compared with regHS and dyHS, the three two-stage methods (mRMR-regHS, mRMR-dyHS, and mRMR-CHS) generally obtain feature subsets with smaller size and achieve higher classification accuracy. This demonstrates the effectiveness of mRMR-CHS.

TABLE III. ACCURACY AND F1 COMPARISONS OF MRMR-CHS AND FILTER MEHTODS

Dataset	w/o		ReliefF		MIM		mRMR		CMIM		JMI		FCBF		mRMR-CHS	
	Acc	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*	*Acc*	*F1*
COLON	75.81	73.32	75.81	72.40	**80.64**	78.50	**80.64**	78.50	77.42	76.56	79.03	76.35	77.42	75.34	**80.64**	78.86
DLBCL	81.82	74.72	85.71	80.46	80.52	74.27	93.51	91.44	89.61	86.02	89.61	86.59	89.61	88.07	**96.10**	95.25
ALLAML	87.50	85.98	93.06	92.31	90.28	89.39	94.44	93.87	**95.83**	95.38	93.06	92.43	93.06	92.29	94.44	93.84
PROSTATE	81.37	81.36	**94.12**	94.15	90.20	90.23	88.23	88.27	87.26	87.41	90.20	90.19	89.22	89.22	92.16	92.15
LEUKEMIA	83.33	83.21	93.06	90.81	93.06	91.71	95.83	95.48	95.83	95.31	95.83	96.72	94.44	94.40	**97.22**	97.79
SRBCT	84.34	85.52	91.57	92.84	98.80	98.89	98.80	98.82	95.18	96.11	98.80	98.89	93.98	95.19	**100.00**	100.00
average	82.36	80.68	88.89	87.16	88.92	87.17	91.91	91.06	90.19	89.46	91.09	90.19	89.62	89.08	**93.43**	92.98

TABLE IV. ACCURACY AND F1 COMPARISONS OF MRMR-CHS AND OTHER HARMONY SEARCH-BASED METHODS

Dataset	regHS				dyHS				mRMR-regHS_				mRMR-dyHS				mRMR-CHS			
	Acc	*F1*	*#gene*	*#std*	*Acc*	*F1*	*#gene*	*#std*	*Acc*	*F1*	*#gene*	*#std*	*Acc*	*F1*	*#gene*	*#std*	*Acc*	*F1*	*#gene*	*#std*
COLON	75.81	72.89	981.10	17.84	69.36	66.20	71.90	4.82	**80.64**	78.50	71.50	3.95	**80.64**	78.86	69.20	5.16	**80.64**	78.86	71.70	7.30
DLBCL	84.42	78.35	3526.1	35.72	83.12	76.19	61.60	3.13	93.51	91.92	60.00	2.45	92.21	91.29	60.60	3.17	**96.10**	95.25	61.20	3.43
ALLAML	87.50	86.02	3566.1	68.44	88.89	87.58	62.40	2.32	**95.83**	95.38	60.30	3.71	94.44	93.87	58.90	2.02	94.44	93.84	61.90	2.77
PROSTATE	81.37	81.36	6300.8	64.76	82.35	82.35	71.60	7.28	90.20	90.19	71.00	6.29	89.22	89.22	74.60	5.87	**92.16**	92.15	72.10	4.07
LEUKEMIA	81.94	80.53	2617.6	31.08	83.33	83.21	62.50	2.68	94.44	94.43	63.20	3.26	**97.22**	97.79	61.60	3.66	**97.22**	97.79	62.80	2.94
SRBCT	86.75	87.78	1140.7	30.05	80.72	82.00	62.00	2.87	96.39	96.96	61.20	1.87	96.39	97.02	59.30	2.21	**100.00**	100.00	60.90	3.11
average	82.97	81.15	3022.07	41.31	81.30	79.59	65.33	3.85	91.83	91.23	64.53	3.59	91.69	91.34	64.03	3.68	**93.43**	92.98	65.10	3.94

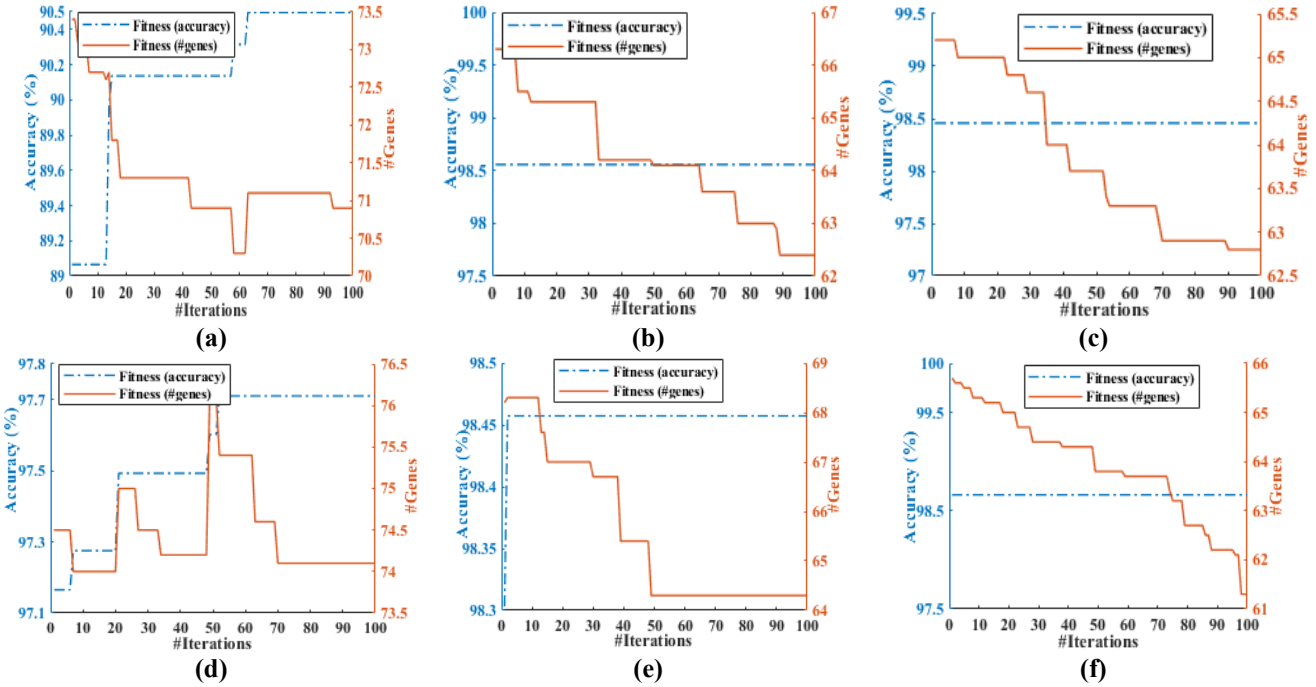

Fig. 2. Convergence curves of the fitness function on each dataset. (a) *COLON*; (b) *DLBCL*; (c) *ALLAML*; (d) *PROSTATE*; (e) *LEUKEMIA*; (f) *SRBCT*.

D. Convergence Analysis

In this section, we investigate the convergence of mRMR-CHS along with the number of improvisations. Fig. 2 presents the curve of the convergence for the two components of the fitness function on each of the six datasets. The X-axis denotes the number of iterations, the left Y-axis refers to the accuracy, and the right Y-axis means the number of selected features. From Fig. 2, we observe that the number of selected features generally decreases and accuracy tends to increase during the iterative process, which leads to the improved fitness value. Second, we see that mRMR-CHS has a fast convergence rate and converges within one-hundred iteration steps, which demonstrates the efficiency of the proposed method.

IV. CONCLUSION

With an aim to select a subset of good features from the high-dimensional small-sample-size microarray data, in this study, we develop a hybrid feature selection method based on the chaotic multi-objective harmony search. Specifically, mRMR is first used to pre-select a small subset of relevant features from the original feature space, and then the chaotic harmony search, which replaces harmony search parameters with the chaotic map during the optimization process, is applied on the reduced feature set to find the optimal feature subset. Finally, extensive comparative experiments against six filter methods (i.e., ReliefF, MIM, mRMR, CMIM, JMI, and FCBF) and four wrapper methods (i.e., regHS, dyHS, mRMR-regHS, and mRMR-dyHS) are conducted on six microarray datasets. The results show that mRMR-CHS achieves higher accuracy than its competitors. In addition, the convergence analysis indicates its efficiency.

REFERENCES

[1] D. G. Beer, S. L. Kardia, C. C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, et al., "Gene-expression profiles predict survival of patients with lung adenocarcinoma," *Nature Medicine*, vol. 8, no. 8, pp. 816-824, 2002.

[2] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 389-422, 2002.

[3] A. Wang, S. Zhao, J. Liu, J. Yang, L. Liu, and G. Chen, "Locality adaptive preserving projections for linear dimensionality reduction," *Expert Systems with Applications*, vol. 151, p. 113352, 2020.

[4] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys*, vol. 50, no. 6, pp. 1-45, 2017.

[5] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, 2005.

[6] A. Wang, N. An, J. Yang, G. Chen, L. Li, and G. Alterovitz, "Wrapper-based gene selection with Markov blanket," *Computers in Biology and Medicine*, vol. 81, pp. 11-23, 2017.

[7] A. K. Shukla, P. Singh, and M. Vardhan, "Gene selection for cancer types classification using novel hybrid metaheuristics approach," *Swarm and Evolutionary Computation*, vol. 54, p. 100661, 2020.

[8] M. Tubishat, N. Idris, L. Shuib, M. AM Abushariah, and S. Mirjalili, "Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Systems with Applications*, vol. 145, p. 113122, 2020.

[9] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60-68, 2001.

[10] L. Coelho, V. Mariani, "Use of chaotic sequences in a biologically inspired algorithm for engineering design optimization," *Expert Systems with Applications*, vol. 34, no. 3, pp. 1905-1918, 2008.

[11] C. Ambroise and G. J. McLachlan, "Selection bias in gene extraction on the basis of microarray gene-expression data," *Proceedings of the National Academy of Sciences*, vol. 99, no. 10, pp. 6562-6566, 2002.

[12] A. Wang, N. An, G. Chen, L. Li, and G. Alterovitz, "Accelerating wrapper-based feature selection with K-nearest-neighbor," *Knowledge Based Systems*, vol. 83, pp. 81-91, 2015.

[13] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method," *Bioinformatics*, vol. 17, no. 12, pp. 1131-1142, 2001.

[14] G. Brown, A. Pocock, M. J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 27-66, 2012.

[15] A. Wang, H. Liu, J. Liu, H. Ding, J. Yang, and G. Chen, "Stable and accurate feature selection from microarray data with ensembled fast correlation based filter," Proc. 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE Press, pp. 2996-2998, Dec. 2020.

[16] T. M. Khoshgoftaar, A. Fazelpour, D. J. Dittman, and A. Napolitano, "Classification performance of three approaches for combining data sampling and gene selection on bioinformatics data," Proc. 2014 IEEE 15th International Conference on Information Reuse and Integration (IRI), IEEE Press, pp. 315-321, Aug. 2014.