A Tunable Particle Swarm Size Optimization Algorithm for Feature Selection

Naresh Mallenahalli¹, T. Hitendra Sarma²
¹Scientist, Software and Database Systems Group, National Remote Sensing Center (ISRO), Hyderabad, Andhra Pradesh, 500 625, India.
e-mail: nareshkumar_m@nrsce.gov.in
² Srinivasa Ramanujan Institute of Technology, Anantapur, Andhra Pradesh, 515 701, India.
e-mail: t.hitendrasarma@gmail.com

Abstract—Feature selection is the process of identifying statistically most relevant features to improve the predictive capabilities of the classifiers. To find the best features subsets, the population based approaches like Particle Swarm Optimization(PSO) and genetic algorithms are being widely employed. However, it is a general observation that not having right set of particles in the swarm may result in sub-optimal solutions, affecting the accuracies of classifiers. To address this issue, we propose a novel tunable swarm size approach to reconfigure the particles in a standard PSO, based on the data sets, in real time. The proposed algorithm is named as Tunable Particle Swarm Size Optimization Algorithm (TPSO). It is a wrapper based approach wherein an Alternating Decision Tree (ADT) classifier is used for identifying influential feature subset, which is further evaluated by a new objective function which integrates the Classification Accuracy (CA) with a modified F-Score, to ensure better classification accuracy over varying population sizes. Experimental studies on bench mark data sets and Wilcoxon statistical test have proved the fact that the proposed algorithm (TPSO) is efficient in identifying optimal feature subsets that improve classification accuracies of base classifiers in comparison to its standalone form.

Index Terms—Feature Selection, Evolutionary Computing, Particle Swarm Optimization, Alternate Decision Trees.

I. INTRODUCTION

In statistical pattern recognition, each pattern represents a real world object described by a set of features (synonymously called as dimensions, here after). More the number of features used, better the description of the object. However, all the features may not be important for the decision making problem on hand. For instance, a student can be described with the features like height, weight, regularity, father name, family income, etc. Now, for the problem on hand, like selecting a student for a basket ball team, the feature like height and weight are highly relevant, whereas the features like father name, family income are irrelevant. The features, regularity and family income are highly relevant to classify whether the student shall be awarded the fellowship or not. Hence, best feature selection for the problem on hand is important for quality decision making.

1This paper is accepted for oral presentation at IEEE Congress on Evolutionary Computation (CEC) - WCCI 2018, Rio de Janerio, Brazil, #18120

Now-a-days, with the development of high-throughput technologies, it is possible to measure hundreds of feature values for each object, which has resulted in large volumes of high dimensional data for analysis. In hyper spectral image analysis, using advanced hyper spectral instruments, hundreds of feature values (each one corresponding to one spectral band) can be measured for each object on earth [1]. In contemporary scientific applications, it is quite often to get such large volumes of high dimensional data, which becomes very challenging problem for analysis [2].

In pattern classification, irrelevant (some times redundant or noisy) features will affect the classification accuracy. It has been proved that, in the presence of large number of features, the learning models become overfit on the training data, which leads to poor generalizability of the trained model, offering a great challenge for pattern classification and prediction problems. Thus, the feature selection process has been considered as a pre-processing step to eliminate irrelevant and redundant features, which is critical for decision making in real world applications [3–7].

The feature selection algorithms have been widely used in many application areas such as genomic analysis [8], text classification [9], information retrieval [10], intrusion detection [11], bioinformatics [12] etc. A comprehensive survey on feature selection methods is published in [13]. Empirical studies on feature selection algorithms for real world problems are presented in [14–17].

Feature selection is an optimization problem which aims to determine an optimal subset of d features out of n features in the input data (d << n), that maximize the classification or prediction accuracy. Performing an exhaustive search to find an optimal subset of d features out of all possible 2n candidate feature subsets, based on some evaluation criterion, is computationally infeasible, and it becomes an NP-hard problem with the increasing n value [13]. Hence, different other search strategies like complete, sequential, random search are explored. However, most of these approaches suffers from local minima problem. Therefore, Evolutionary Computation (EC) techniques, which ensures global optimum or near global optimum, such as Genetic Algorithms(GAs) [13], Genetic Programming (GP) and Particle Swarm Optimization (PSO),
were used in many feature selection problems. As stated in
[19], PSO is simple to understand and easy to implement than
GP and GAs and able to handle optimization problems with
multiple local optima reasonably well, it requires less number
of parameters and can converge more quickly. However, the
efficiency of PSO depends on various input parameters that
are to be tuned properly [20], [21]. More detailed study on
PSO and its improvements is presented in [22].

In the standard PSO, the swarm size is an important par-
eter, where very small swarm size will lead to local minima,
while large swarm size would slow down the algorithm [23].
To address this issue, in the present work, we intend to vary
the population sizes of the particles in standard PSO based
on the data sets in real time. A new objective function has
been developed which integrates the accuracy of the classifier
with the modified F-Score. Finally, we propose a new PSO
search method for feature selection using tunable swarm size configuration. The efficiency of the proposed method is
compared with other popular contemporary feature selection
methods.

This paper is organized as follows. Section II presents
the brief review of the existing methods for feature selec-
tion. In Section III we briefly outlined the standard PSO
methodology and presented the motivation for the tunable
swarm size configuration in the present work. Section IV
outlines the Alternating Decision Tree classifier, which is
used along with the standard PSO for feature subset selection.
The proposed Tunable Particle Swarm Size Optimization (TPSO)
algorithm is presented in Section V. The experiments and
results are presented in Section VI. Conclusions and discussion
are presented in Section VII.

II. FEATURE SELECTION METHODS

In literature, the feature selection methods are broadly
classified into three categories viz., filter, wrapper and en-
bedded methods. Filter methods select the feature based on
the given data, irrespective of the classifier. In the wrapper
model, feature selection will be done based on the feedback
of the predefined learning model. Wrapper based methods will
find better and optimal feature subsets with high accuracy,
as they are considering the feedback of the learning model,
but it requires expensive computation. However, it is proved
that filters have better generalization capabilities than wrapper
based ones [8].

Algorithms with embedded models such as C4.5 [24] and
least angle regression (LARS) [25], the variable or feature
selection process is incorporated as part of the training process,
and the relevance of the selected feature is analyzed by the
objective function of the learning model under consideration.
Both filter and embedded approaches may result a subset of
selected features or the weights that represent the relevance or
importance of all features.

Some feature selection methods compute the ranks of all
features using some ranking criterion, such methods are simple
and computationally efficient. These rank based methods are
more robust against over fitting, resulting more bias with less
variance [4], [26]. Further, the statistical approaches such as T-
Statistics, F-Statistics, Chi-square test etc., have been explored
significantly in the literature [27], [28]. A few other feature
selection approaches are based on the concepts of information
theory such as information gain [29], mutual information [4],
and entropy-based measure [6]. Machine learning tech-
niques including evolutionary algorithms, SVMs, Decision
Trees etc., are also been used for feature selection. [31–
33]. More recently, the evolutionary Computing techniques
such as such as Genetic Algorithms(GAs) and Particle Swarm
Optimization (PSO) are being used popularly used for feature
selection. Bing Xue et.al., explored the performance of PSO
and various other improvements in [22]. PSO is widely used
for Feature Selection on High-dimensional Datasets [34]. A
good survey on novel population topologies for improving the
performances of population-based optimization algorithms for
solving single objective optimization, multiobjective optimiza-
tion and other classes of optimization problems is presented in
[35].

This paper presents an improvement over the standard
PSO, which is a wrapper based approach to improve the
classification accuracy with reduced number of features.

III. STANDARD PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization imitates the movement of a
flock of birds, where each bird has its own intelligence to
find the best direction to move and to reach the destination as
a whole. In standard PSO, each single candidate solution is
considered as a particle in the search space. For each particle,
there is a fitness value, computed using a fitness function to
be optimized, and velocity, which determine the movement
of the particles. During movement, each particle updates its
position based on its previous position, velocity and as well
as considering the positions of neighbouring particle.

The standard PSO starts with a randomly initialized popu-
lation (particles) of size N. Each particle P_i is identified as
a point in the d dimensional space X_i = (X_{i1}, X_{i2}, . . . , X_{id}).
pbest represents the fitness values of the best positions of the
particles given by F_i = (f_{i1}, f_{i2}, . . . , f_{id}). gbest represents
the index of the particle that has the best fitness value in
the swarm. The velocity of a particle i is represented by
V_i = (v_{i1}, v_{i2}, . . . , v_{id}).

The iterative approach starts with an initial random solutions
(particles in initial swarm). In each iteration, for each particle,
the velocity and the position are updated using the following
equations:

\[ x_{ij}^t = x_{ij}^{t-1} + v_{ij}^t \]  \hspace{1cm} (1)
\[ v_{ij}^t = w \cdot v_{ij}^{t-1} + \eta_1 \cdot r1(*) \cdot (p_{ij} - x_{ij}) + \eta_2 \cdot r2(*) \cdot (g_{ij} - x_{ij}) \]  \hspace{1cm} (2)

where \( j = 1, \ldots, d \), \( w \) is a positive linear function of time
which updates according to the generation iteration. The \( \eta_1 \)
and \( \eta_2 \) represent the acceleration terms that pull the particles
towards pbest and gbest. The \( r1(*) \) and \( r2(*) \) are random
number generation functions, which generates random values

This paper is accepted for oral presentation at IEEE Congress on Evolutionary Computation (CEC) - WCCI 2018, Rio de
2 Janerio, Brazil , #18120
that are uniformly distributed in $[0, 1]$. The terms $p_{ij}$ and $p_{gj}$ represent the $p_{best}$ and $g_{best}$ in the $j^{th}$ dimension respectively. The velocities of the particles are bounded by a maximum limit $V_{max}$. If $V_{max}$ is too small then it may end up with a local optima, and if the $V_{max}$ is too large then the particles may fly beyond the good solutions.

The swarm size is a critical parameter in this standard PSO algorithm wherein very few particles will make the algorithm to get stuck at the local optima, while too many particles would slow down the algorithm [23]. It is the key factor that has motivated the present research work.

In this paper, we propose a new particle swarm optimization search for feature subset selection using tunable swarm size configuration, which is explained in Section IV.

IV. ALTERNATING DECISION TREES (ADT)

Alternating Decision Trees (ADT) are often considered as generalization of conventional decision trees [36]. ADT generates the decision rules based on majority voting taking all simple rules into account. It consists of decision nodes and prediction nodes. The prediction nodes contain a numeric value having a positive or negative sign, and the decision node specify a condition. The decision nodes will be splitting nodes whereas the prediction nodes are either root or leaf nodes. An instance is classified by traversing from the root by following all paths where all the decision nodes are true. A positive sum of all prediction nodes that are been traversed implies the membership of one class and the negative sum implies the membership of other class. Empirical studies proved that, under some favourable conditions ADTs are more robust than the conventional decision trees, C4.5 and J48 [37].

V. TUNABLE PARTICLE SWARM SIZE OPTIMIZATION ALGORITHM (TPSO)

In this section we present our new algorithm called Tunable Particle Swarm Size Optimization Algorithm (TPSO) which will find the best initial swarm size for the given data to overcome the local minima problem [23].

The data set is split into testing and training folds using a stratified $k$ fold cross validation procedure. For each of the training data sets we first initialize swarm size and then select the features using the standard PSO and Alternating Decision Tree (ADT). We then compute the test accuracy using the features subset identified in the previous step and ADT classifier. A new feature score which measures the discrimination between features having two sets of numbers categorical or numeric with respect to the decision attribute is then computed following the procedure in Section V-A.

A. New Feature Discrimination Score

Consider a given the data set $X_{ij}$, for $j = 1, 2, \ldots , n - 1$, $n = 1, 2, \ldots , m$ having $n$ rows and $m$ features where the last feature is the decision class. If the decision class is binary then $n^+$ and $n^−$ denote positive and negative instances respectively. In [38] a feature discriminatory score using mean of the attribute values is computed. In our approach instead of mean we employ median as it is the best representative the central tendency of data sets with skewed distribution. We define the feature score of the $i^{th}$ feature as:

$$F(i) = \frac{V_1}{V_2}$$  (3)

$$V_1 = abs((\tilde{x}_i^+ - \tilde{x}_i^n)^2 + (\tilde{x}_i^- - \tilde{x}_i^n)^2)$$  (4)

$$V_2 = \frac{1}{n^n - 1} \sum_{k=1}^{n^n} (x_{k,i}^+ - \tilde{x}_i^n)^2 + \frac{1}{n^n - 1} \sum_{k=1}^{n^n} (x_{k,i}^- - \tilde{x}_i^n)^2$$  (5)

where $\tilde{x}_i^+$ denotes the median of the values in the $i^{th}$ attribute corresponding to the positive decision class, $\tilde{x}_i^−$ denotes the median of the attribute values corresponding to the negative decision class, $\tilde{x}_i^+$ denotes the median of all the values of the $i^{th}$ attribute, $x_{k,i}^+$ is the median value of the $i^{th}$ feature of the $k^{th}$ positive instance and $x_{k,i}^−$ is the median value of the $i^{th}$ feature of the $k^{th}$ negative instance.

B. Fitness function

We develop a new fitness function to evaluate the effectiveness of the feature subsets as mentioned below.

$$V_i = 0.5 * A + 0.5 * (M_1/M_2)$$  (6)

where $A$ is the accuracy obtained using ADT, $M_1$ is the sum of the discriminatory scores of the features in the reduced feature subset, that is $M_1 = \sum_{i=1}^{r} F(i)$ where $r < m$, $M_2$ is the sum of the discriminatory scores of all the features in the data set, that is $M_2 = \sum_{i=1}^{m} F(i)$. We assume the condition that $M_2 > 0$ as $\sum_{i=1}^{m} F(i) > 0$.

In the proposed algorithm we perform a stratified $k$ fold cross validation and split the data set into ten training and test data sets. For each training data set we extract the feature subset using standard PSO and ADT classifier. We then compute the feature discrimination score using the formula [3]

We compare the new feature score with the previous scores and the algorithm increases the particle population size in the standard PSO by a factor of one till a local maximum is found. To obtain the local maximum point we first obtain the first and second derivative of number of particles in the iteration $i$ (say $y$) and the feature discriminatory score $V_i$ (say $x$). The local maxima is computed as given in Equation 7. The loop is terminated when the conditions in Equation 7 are met.

$$\frac{dy}{dx} > \frac{dy}{dx} i-1 \& \&$$

$$\frac{dy}{dx} > \frac{dy}{dx} i \& \& \frac{d^2 y}{dx^2} < 0$$  (7)

This paper is accepted for oral presentation at IEEE Congress on Evolutionary Computation (CEC) - WCCI 2018, Rio de Janeiro, Brazil, #18120
Algorithm 1 New Feature Selection (NFS) Algorithm

**Input:** Data sets for the purpose of decision making $S(n,m)$ where $n$ and $m$ are number of records and attributes respectively.

**Output:** Mean and standard deviation of the number of features selected and the classification accuracy $FM$, $FS$, $AM$, $AS$

1) Identify and collect all records in a data set $S$.
2) Split the data set into training and testing sets using a stratified $k$ fold cross validation procedure. Denote each training and testing data set by $T_k$ and $R_k$ respectively.
3) For each $k$
   a) initialize the number of particles in PSO as $N=5$, the fitness value as $V_0 = 0$ and $i = 0$.
   i) Extract the optimal feature subset from $T_k$ using a wrapper based approach with PSO search for identifying feature subsets using number of particles as $N$ and alternating decision tree for its evaluation.
   ii) Generate new training and testing data sets using the features identified in Step 3(a) from $T_k$ and $R_k$ respectively. Designate these sets by $P_i$ and $Q_i$ respectively.
   iii) Build the ADT using the training data set $P_i$ and obtain the test accuracy using testing data set $Q_i$. Designate this accuracy by $B_i$.
   iv) Compute the feature scores of data sets $P_i$ and $T_k$ respectively. Designate the scores by $M_1$ and $M_2$ respectively.
   v) Compute the fitness value as $V_i = 0.5 \times A + 0.5 \times (M_1/M_2)$.
   vi) Update $N$ as $N + 1$ and $i$ as $i + 1$.
   vii) check for local maxima using Equation 7 and exit the loop if conditions are met.
   b) Obtain the feature subset using the particles obtained in set above in PSO feature selection using ADT.
   c) Obtain the test accuracy $A_k$ by building an ADT using the features obtained from above and designate the accuracy of fold $k$ and $F_k = \#D_i$.
4) Repeat the Steps (3)-(i) to Step (3)-3c for each fold.
5) Designate $FM$, $FS$, $AM$, $AS$ as the mean and standard deviation of $F_k$ and $A_k$ respectively.
6) RETURN $FM$, $FS$, $AM$, $AS$.
7) END.

C. Validation

To evaluate the performance of the proposed Tunable Particle Swarm Size Optimization Algorithm, we first obtain the feature subset corresponding to the number of particles found using the above procedure. We then train an ADT using the features identified in the previous step and then compute test accuracies on the test data set of the corresponding fold. The procedure is repeated for all the ten folds and the average accuracy is computed. The above procedure is given as Algorithm 1.

VI. EXPERIMENTS AND RESULTS

We have conducted experiments on benchmark data sets obtained from University of California Irvin (UCI) data repository StatLog project, Keel and Bangor data repositories (https://www.bangor.ac.uk/). The performance of the proposed algorithm TPSO is compared with standard PSO and GA with alternating decision tree classifier. The description of the data sets are given in the Table I.

In the present methodology we employ a stratified $k$-fold cross validation ($k=10$) procedure. The folds are selected so that the mean response value is approximately equal in all the folds. In case of a dichotomous classification, this means that each fold has roughly the same proportions of the two types of class labels. Table I provides details of the default necessary parameters of GA and PSO in the current experimental study.

A. Computational Complexity and Scalability

The computational complexity is a measure of the performance of the algorithm. For each data set having $n$ attributes and $m$ records, we select only those subset of records $m_1 \leq m$, in which missing values are present. The distances are computed for all attributes $n$ excluding the decision attribute. So, the time complexity for computing the distance would be $O(m_1 \times (n-1))$. The time complexity for selecting the nearest records is of order $O(m_1)$. For computing the frequency of occurrences for nominal attributes and average for numeric attributes the time taken would be of the order $O(m_1)$. In case of the proposed method, let $O(p)$ be the time complexity of wrapper based feature sub set identification using standard PSO and ADT. For $F$ folds the complexity would be $F \times O(p)$. For $I$ changes in the swarm size the time complexity of feature selection step would be $F \times O(p) \times I$. Therefore, for a given data set with $k$-fold cross validation having $n$ attributes and $m$ records, the time complexity of TPSO would be $k \times (O(m_1 \times (n-1)) \times m) + 2 \times O(m_1) + F \times O(p) \times I$ which is asymptotically linear.

A plot between the varying sizes of the data sets and the time taken for processing by the proposed algorithm (TPSO) is shown in Fig. I. Also, we employed a linear regression on our results and obtained the relation between the time taken (T) and the data size (D) as $T = 0.0159D + 0.2464, \alpha = 0.05, p < 0.05, r^2 = 0.78$.

The presence of the linear trend between the time taken and the varying database sizes ensures the numerical scalability of the performance of TPSO in terms of asymptotic linearity.
### TABLE I
**DATA SETS**

| Name                      | Source       | Number of Features | Number of Classes | Number of Records |
|---------------------------|--------------|--------------------|-------------------|-------------------|
| Australian (AUS)          | UCI Statlog  | 14                 | 2                 | 690               |
| Contraction (CON)         | Bangor       | 27                 | 2                 | 98                |
| German (GER)              | Keel         | 20                 | 2                 | 1000              |
| Heart (HR)                | UCI Statlog  | 13                 | 2                 | 270               |
| Ionosphere (ION)          | UCI          | 33                 | 2                 | 351               |
| Laryngeal (LAR)           | Bangor       | 16                 | 2                 | 213               |
| Respiratory distress syndrome (RDS) | Bangor       | 17                 | 2                 | 85                |
| Sonar (SNR)               | UCI          | 60                 | 2                 | 208               |
| WDBC (WBD)                | Keel         | 30                 | 2                 | 569               |
| Weaning (WEA)             | Bangor       | 17                 | 2                 | 302               |

### TABLE II
**PARAMETER SETUP**

| Name       | Parameters                                      |
|------------|-------------------------------------------------|
| GA         | Crossover=1.0, mutation probability=0.001        |
| PSO        | initial number of particles Z=50, iterations G=100, cognitive factor $c_1=2$, social factor $c_2=2$ |

Fig. 1. Computational complexity of the TPSO algorithm

### B. Performance Comparison on Benchmark Data sets

Firstly, we compared the accuracy of the proposed TPSO method with accuracies of ADT classifier without employing any feature selection. The results are tabulated in Table III.

Later, we considered GA and standard PSO algorithms for feature subset selection and ADT classifier as wrapper for feature evaluation. A comparison of the performances of TPSO with GA+ADT and Standard PSO+ADT methods on benchmark data sets is shown in Table IV.

The mean and the standard deviation of the number of features selected for $k$ folds of the cross validation procedure is shown Table V.

From Table V it can be observed that the TPSO methodology has rendered higher accuracies using less than $50\%$ of the original set of attributes.

To substantiate the improvement in classification accuracy using TPSO methodology a statistical test based on Wilcoxon method is employed and the results are presented in Table VI.

From the Table VI we infer that TPSO is superior to the standard PSO feature selection method with positive rank sum of 55, $p < 0.001$ and $\alpha = 0.05$ significance. The TPSO method indicating a remarkable performance when compared with GA feature selection with a positive rank sum of 54, $p < 0.003$ and $\alpha = 0.05$.

### VII. Conclusions and Discussion

In this paper, we have discussed the issues related to high dimensionality of the data sets and feature selection as a solution to the curse of dimensionality. The feature selection methods such as filter and wrapper have been discussed. Particle Swarm Optimization (PSO) is a population based optimization technique, which has been proved to get optimal feature subset provided the necessary input parameters are properly tuned. Particle swarm size is the critical parameter in standard PSO. To address this issue, we proposed a novel tunable swarm size configuration approach to find the population size of the particles based on the data sets in real time. The proposed algorithm is named as Tunable Particle Swarm Size Optimization Algorithm (TPSO). A new fitness function has been developed which integrates the accuracy of the classifier with the modified F-Score. Empirically, we compare the performance of our new algorithm with other state-of-the-art classifiers on benchmark data sets obtained from UCI, Keel and Bangor data repositories. Wilcoxon statistical test confirmed the fact that the proposed algorithm has improved the classification accuracies in comparison to other methods.

### References

[1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[2] J. Hua, W. D. Tembe, and E. R. Dougherty, “Performance of feature-selection methods in the classification of high-dimension data,” *Pattern Recognition*, vol. 42, no. 3, pp. 409–424, 2009.

This paper is accepted for oral presentation at IEEE Congress on Evolutionary Computation (CEC) - WCCI 2018, Rio de Janeiro, Brazil, #18120
### Table III

| Dataset | TPSO       | ADT       |
|---------|------------|-----------|
| 87.4 ± 3.32 | 84.35 ± 3.23 |
| 87.89 ± 10.31 | 84.78 ± 12.81 |
| 74.4 ± 3.75 | 72.8 ± 4.59 |
| 83.3 ± 7.46 | 78.89 ± 10.48 |
| 94.86 ± 3.24 | 90.02 ± 4.92 |
| 86.86 ± 6.91 | 82.23 ± 9.37 |
| 90.56 ± 11.37 | 90.56 ± 11.37 |
| 86.95 ± 8.00 | 83.05 ± 9.25 |
| 97.19 ± 3.12 | 98.38 ± 3.86 |
| 97.19 ± 3.12 | 98.38 ± 3.86 |
| 81.42 ± 3.38 | 81.44 ± 3.38 |

### Table IV

| Dataset | TPSO | GA+ADT | Standard PSO+ADT |
|---------|------|--------|------------------|
| AUS     | 87.54 ± 3.82 | 85.51 ± 3.86 | 84.49 ± 2.65 |
| CON     | 87.89 ± 10.31 | 77.78 ± 14.56 | 80.78 ± 9.91 |
| GER     | 74.4 ± 3.75 | 89.8 ± 4.39 | 70.4 ± 5.64 |
| HRT     | 83.33 ± 7.46 | 83.7 ± 7.24 | 76.67 ± 9.08 |
| ION     | 94.86 ± 3.24 | 89.75 ± 6.74 | 92.01 ± 3.77 |
| LAR     | 86.86 ± 6.91 | 79.81 ± 7.62 | 79.83 ± 9.19 |
| RDS     | 90.56 ± 11.37 | 90.56 ± 8.23 | 89.31 ± 10.9 |
| SNR     | 86.95 ± 8.00 | 81.19 ± 11.33 | 81.12 ± 10.10 |
| WBD     | 97.19 ± 3.12 | 95.26 ± 3.7 | 95.25 ± 3.42 |
| WEA     | 87.42 ± 3.38 | 84.78 ± 3.81 | 82.76 ± 5.88 |

### Table V

| Dataset | Original | TPSO | GA+ADT | Standard PSO+ADT |
|---------|----------|------|--------|------------------|
| AUS     | 14       | 5.8 ± 2.20 | 5.3 ± 1.71 | 2.8 ± 2.78 |
| CON     | 27       | 13.2 ± 3.12 | 9.5 ± 2.46 | 11.9 ± 4.38 |
| GER     | 20       | 10 ± 2.31 | 5.8 ± 2.03 | 5.5 ± 2.68 |
| HRT     | 53       | 9.3 ± 1.49 | 4.3 ± 2.11 | 9.3 ± 1.16 |
| ION     | 34       | 13.8 ± 4.57 | 12.6 ± 2.63 | 11.2 ± 3.88 |
| LAR     | 56       | 4.9 ± 2.18 | 5.4 ± 2.12 | 6.8 ± 2.1 |
| RDS     | 100      | 10.0 ± 2.26 | 6.4 ± 1.35 | 9.3 ± 1.49 |
| SNR     | 190      | 33.8 ± 10.21 | 28.6 ± 4.65 | 35.0 ± 9.93 |
| WBD     | 150      | 13.4 ± 3.06 | 9.4 ± 2.84 | 13.0 ± 2.36 |
| WEA     | 70       | 9.6 ± 2.17 | 8.3 ± 1.25 | 9.7 ± 1.89 |

### Table VI

Wilcoxon sign rank statistics for matched pairs comparing TPSO with other feature selection methods

| Method  | Rank Sums (+, -) | Test Statistics | Critical Value | p-value |
|---------|------------------|-----------------|---------------|---------|
| GA+ADT  | 54.0, 1.0        | 0.0             | 9.0           | 0.001   |
| PSO+ADT | 55.0, 0.0        | 0.0             | 9.0           | 0.001   |

[3] H. Liu, J. Li, and L. Wong, “A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns,” *Genome Informatics Series*, pp. 51–60, 2002.
[4] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” *Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.
[5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, pp. 389–422, 2002.
[6] H. Liu, E. R. Dougherty, J. G. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons, Z. Zhao, L. Yu, and G. Forman, “Evolving feature selection,” *IEEE Intelligent Systems*, vol. 20, pp. 64–76, 2005.
[7] H. Yang, Q. Du, and G. Chen, “Particle swarm optimization-based hyperspectral dimensionality reduction for urban land cover classification,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 544–554, 2012.
[8] I. n. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza, “Filter versus wrapper gene selection approaches in dna microarray domains.” *Artificial intelligence in medicine*, vol. 31(2), pp. 91–103, 2004.
[9] F. George, “An extensive empirical study of feature selection metrics for text classification,” *Journal of Machine Learning Research*, vol. 3, pp. 1289–1305, 2003.
[10] D. Swets and J. Weng, “Efficient content-based image retrieval using automatic feature selection,” in *IEEE International Symposium On Computer Vision*, 1995, pp. 85–90.

This paper is accepted for oral presentation at IEEE Congress on Evolutionary Computation (CEC) - WCCI 2018, Rio de Janeiro, Brazil, #18120
