CHAPTER 2

WRAPPER BASED ATTRIBUTE SELECTION USING
BPSO WITH GREEDY RESET AND LOCALIZED
RANDOM MISTAKES

Wrapper based attribute selection employs a search technique to select an attribute subset on traversing the problem space, which optimizes the criterion, $\text{FoM}$. $\text{FoM}$ is evaluated using a classifier with selected attributes as input. This chapter presents a BPSO based attribute selection methodology to select an optimal attribute subset that achieves improved predictive performance with minimal attributes than that with complete set of attributes. Nevertheless, particle swarm is a partial optimizer (Talukder 2011; Chuang et al. 2008; Zhang et al. 2014) resulting in pre-mature convergence to a sub-optimal point in the problem space. Greedy reset with localized random mistakes is devised, to divert the trapped particles to explore untried areas in the problem space.

2.1 PARTICLE SWARM OPTIMIZATION

Particle swarm optimization works with solutions discovered by particles in the swarm on traversing the problem space. PSO algorithm is outlined in Figure 2.1. Each particle in the swarm represents a solution in the problem space and initialized with random velocity and a position in the problem space. Velocity directs the flight of the particle in the problem space in search of an optimal solution. The particles, then move from one position to another in the problem space according to its current velocity. In the course
of search, velocity of the particle needs to be updated such that the particles move towards the target.

Velocity of each particle is updated in every run on collectively considering the past velocity of the particle and the two best values defined below:

i) Best position confronted by each particle in the past, Ownbest

ii) Best position confronted by the swarm ie., best position achieved by any particle in the swarm, Swarmbest.

```plaintext
begin
    Define problem space and FoM
    Generate a swarm of N particles with initial velocity and position
    while (maximum runs is not met) do
        for each particle
            Evaluate FoM
            Record Ownbest and Swarmbest
            Update velocity and position of the particle
        endfor
    endwhile
end
```

**Figure 2.1 Particle Swarm Optimization Algorithm**
Thus, particle’s search is directed based on swarm intelligence (collective intelligence of self organized groups) and social interaction within the swarm. Finally, all the particles in the swarm span towards an optimum solution in the problem space by updating its velocity and position.

2.2 **BINARY PARTICLE SWARM OPTIMIZATION**

Attribute selection is a combinatorial problem that involves inclusion or exclusion of attributes to identify an optimal subset of highly predictive attributes for superior classification performance. Thus, the problem of attribute selection demands a large Hamming space with all combination of attributes as solution vectors on it. A BPSO is more appropriate to explore and exploit the discrete problem space than a PSO that best combs a continuous problem space (Kennedy & Eberhart 1997). Pseudocode of wrapper based attribute selection using BPSO is given in Figure 2.2.

Position vector of $N$ particles in the swarm with $D$ dimensions each represent the status (presence or absence) of attributes in the subset. Initial position vector of $N$ particles is selected randomly from the problem space i.e., each particle is assigned any one of the possible combination of attribute subset. Each particle is represented with index $i, i=1, 2, \ldots N$.

Position of particles in the swarm is denoted as$(X_1^{run}, X_2^{run}, \ldots X_N^{run})$.

$$X_i^{run} = (x_{i1}^{run}, x_{i2}^{run}, \ldots x_{ij}^{run}, \ldots x_{iD}^{run})$$ represent position vector of particle, $i$ with dimension $j, j=1, 2, \ldots D$ in the current run.

$$x_{ij}^{run} = \begin{cases} 
1 & \text{Presence of attribute } j \text{ in the subset} \\
0 & \text{Absence of attribute } j \text{ in the subset}
\end{cases}$$

(2.1)
begin
    Define a $D$ dimensional Hamming space, $R^D$
    Represent all possible combination of $D$ attributes as elements of $R^D$
    Randomly place $N$ particles in the problem space, $R^D$
    Initialize velocity of each particle using Equation (2.2)
    while (maximum runs is not met) do
        for each particle
            Evaluate $FoM$ using Equation (1.5)
            Calculate $Ownbest$ and $Swarmbest$
            for each dimension of particle
                Update velocity using Equation (2.3)
                Update position using Equation (2.7)
            end for
        end for
    end while
    Return $Swarmbest$ as the optimal attribute subset
end

Figure 2.2 Wrapper based Attribute Selection using BPSO

Velocity of the particle decides the change in direction of particle’s flight towards the potential solution. It represents step size that each particle has to take to comb the problem space in search for an optimal solution. Initial velocity of each particle is calculated using Equation (2.2).

$$v_{ij}^{un} = v_{min} + (v_{max} - v_{min}) * rand (0, 1)$$  \hspace{1cm} (2.2)
\((V_1^{run}, V_2^{run}, \ldots, V_N^{run})\) denote velocity of each particle in the swarm.

\[ V_i^{run} = (v_{i1}^{run}, v_{i2}^{run}, \ldots, v_{ij}^{run}, \ldots, v_{iD}^{run}) \]
marks velocity of \(i^{th}\) particle at \(j^{th}\) dimension in the current \(run\). \(v_{min}\) and \(v_{max}\) represent minimum and maximum allowable velocity and \(N\) is the number of particles in the swarm with \(D\) dimensions each. \textbf{rand} operator generates a random number \(\in (0,1)\)

On initializing position and velocity, particles span through the search space to adjust their velocity and position in accordance with their own knowledge \((\text{Ownbest})\) and that of the swarm \((\text{Swarmbest})\) through social interaction.

Each particle in the swarm demands neighbours for social interaction. Neighbourhood of each particle for social interaction is the entire swarm and it is represented by a star topology (Talbi 2009) with a fully connected network. Each particle interacts with every other particle in the swarm, as a result star topology aids in faster convergence (Engelbrecht 2007).

In the process of search, step size of each particle required for traversing the problem space towards the target is decided in every \(run\). It is an indication of how far the individual data is from the target. If the particle is farther from the target, velocity of particle’s flight is increased to reach the target. It is derived on considering the memory of flight direction in the immediate past, memory of the position that was best so far for the particle and that of the entire swarm. Velocity updation in each dimension of the particle is calculated using Equation (2.3)

\[
\begin{align*}
v_{ij}^{run+1} &= \omega v_{ij}^{run} + c_1 r_{ij}^{run} [ownbest_{ij}^{run} - x_{ij}^{run}] + c_2 r_{2ij}^{run} [Swarmbest_j - x_{ij}^{run}] \\
(2.3)
\end{align*}
\]
where $v_{ij}^{run}$ and $v_{ij}^{run+1}$ marks velocity of $i^{th}$ particle in $j^{th}$ dimension of current and next run respectively. $\omega$ is the inertia weight. $C_1$ and $C_2$ are the acceleration constants, $r_{ij}^{run}$ and $r_{2j}^{run} \in (0,1)$ are random numbers in $j^{th}$ dimension of current run. $x_{ij}^{run}$ represent position vector of particle in the current run. $ownbest_{ij}^{run}$ is the personal best of $i^{th}$ particle in $j^{th}$ dimension obtained on search so far. $Ownbest$ position of each particle, $i$, in the swarm is denoted by

$$Ownbest_{i}^{run} = (ownbest_{i1}^{run}, ownbest_{i2}^{run} \ldots ownbest_{ij}^{run} \ldots ownbest_{iD}^{run})$$

$$ownbest_{ij}^{run} = \begin{cases} 
1 & \text{Presence of attribute } j \text{ in } Ownbest_{i}^{run} \\
0 & \text{Absence of attribute } j \text{ in } Ownbest_{i}^{run} 
\end{cases}$$

(2.4)

$Swarmbest$ is the best solution found so far during search by the entire swarm of particles. It is represented as

$$Swarmbest = (swarmbest_1 \ldots swarmbest_j \ldots swarmbest_D)$$

$$swarmbest_j = \begin{cases} 
1 & \text{Presence of attribute } j \text{ in } swarmbest \\
0 & \text{Absence of attribute } j \text{ in } swarmbest 
\end{cases}$$

(2.5)

Velocity update uses acceleration coefficients, $C_1$ and $C_2$ to measure the confidence that a particle has on itself and on its neighbours in the swarm respectively (Engelbrecht 2007). To balance both exploration and exploitation, $C_1$ is kept equal to $C_2$ such that the particles move towards average of $Swarmbest$ and its $Ownbest$ (Engelbrecht 2007;
Van den bergh & Engelbrecht 2006). Both the coefficients along with \( r_1 \) and \( r_2 \) maintain stochasticity during velocity updation of particles.

Velocity update is inferred as the change in the probability of finding the particle in one state or another (Kennedy & Eberhart 1997). Hence velocity is mapped to an interval \((0,1)\) by using monotonically increasing sigmoid function as in Equation (2.6)

\[
s_{ij}^{\text{run}} = \frac{1}{1 + e^{-v_{ij}^{\text{run}+1}}}
\]  

(2.6)

where \( s_{ij}^{\text{run}} \) denote normalized velocity of \( i^{th} \) particle in \( j^{th} \) dimension of current \( \text{run} \) on the updated velocity, \( v_{ij}^{\text{run}+1} \).

Position of particle in each dimension is updated using Equation (2.7)

\[
x_{ij}^{\text{run}+1} = \begin{cases} 
1 & u_{ij}^{\text{run}} < s_{ij}^{\text{run}} \\
0 & u_{ij}^{\text{run}} \geq s_{ij}^{\text{run}} 
\end{cases}
\]  

(2.7)

where \( u_{ij}^{\text{run}} \) is a random number in the interval \((0,1)\) and \( x_{ij}^{\text{run}+1} \) indicates position (inclusion or exclusion of attribute) of \( i^{th} \) particle in \( j^{th} \) dimension in the next \( \text{run} \).

Finally, \textit{Swarmbest} and \textit{Ownbest} solution vectors are restructured on comparing with \textit{FoM} of current and past potential solutions obtained during the search such that the particles are directed towards optimum point in the problem space. This process is iteratively repeated till it reaches a pre-fixed maximum number of \( \text{runs} \). On termination, \textit{Swarmbest} is the optimal solution obtained on executing BPSO as the search strategy.
2.3 LITERATURE SURVEY ON BPSO FOR ATTRIBUTE SELECTION

Many researchers have exploited BPSO and SVM in implementing wrapper based attribute selection methodology. Authors have also proposed variants in particle swarm to improve the efficiency in selecting highly predictive attributes.

Agrafiotis & Cedeno (2002) aims to select an optimal attribute subset using PSO and Neural Network (NN). This methodology requires the user to specify the size of attribute subset for selection which is rather difficult without prior knowledge about the attribute space and is problem specific.

Azevedo et al. (2007) employed BPSO and SVM to build a wrapper based attribute selection methodology for identifying keystroke dynamic systems. Tu et al. (2007) designed an attribute selection technique using particle swarm optimizer and SVM with one-versus-all method to classify multi class problems.

Lin et al. (2008) developed a wrapper based attribute selection methodology using BPSO and SVM. The proposed methodology traverses the problem space to select an optimal attribute subset simultaneously optimizing the parameters of SVM. Unlike Lin et al. (2008), Huang & Dun (2008) applied continuous valued PSO for optimizing parameters in SVM and BPSO for optimizing attribute subset search.

Unler & Murat (2010) introduced an adaptive selection strategy to select an attribute subset based on both its likelihood and its influence on other attributes already added to the subset. This strategy shows superior performance to Tabu search and scatter search algorithm. The authors also
state that the efficiency of adaptive selection strategy and the quality of solution can be improved by relaxing the restriction on the size of attributes considered for adding to the subset.

A multi-swarm BPSO methodology, to simultaneously optimize parameters in SVM and the search for an attribute subset with high predictive accuracy is proposed by Liu et al. (2011). However, this methodology is computationally expensive as it employs more particles in the swarm for search with complex communication between different sub swarms.

In the process of wrapper based attribute selection using BPSO, position of each particle is a binary string in Hamming space requiring no boundary conditions (Xu & Rahmat-Samii 2007). Hence, the problem of divergence does not occur; however, the problem of pre-mature convergence occurs in two cases:

**Case (i):**

When *Swarmbest* in itself is locked in a sub-optimal solution in the problem space- *Swarmbest stagnation*

**Case (ii):**

New areas are untried when velocity reaches maximum or minimum (stagnates) as it corresponds to bit position left unchanged at one and zero respectively- *Velocity stagnation.*

### 2.3.1 *Swarmbest* Stagnation

Each particle of the population, in search of an optimal solution, adjusts its position depending on its personal best and the best solution chosen on social interaction with neighbours. In conventional BPSO, *Swarmbest* is
updated only on achieving a better solution than the previous \textit{Swarmbest}. If the \textit{Swarmbest} in itself gets stagnated in local optima, particle’s search area is restricted around the \textit{Swarmbest} in the problem space. Particles don’t explore the entire problem space and limits its search around the local optima. Such a solution (attribute subset) may not yield superior predictive results during classification.

In addition, BPSO has a tendency to converge rapidly during initial search and retards the convergence rate quite often. As a consequence, particles get trapped in local optima (Stacey \textit{et al.} 2003; Zhan \textit{et al.} 2009). Also, Alba \textit{et al.} (2007); Yuan & Zhao (2007); Chuang \textit{et al.} (2008); Zhang \textit{et al.} (2014) reported pre-mature convergence in BPSO.

To avoid particles getting trapped in local optima, \textit{Swarmbest} is monitored in each \textit{run}. On stagnation, if \textit{Swarmbest} remains unchanged for a prefixed number of \textit{runs}, \textit{Swarmbest} is reset. Authors have employed operators such as selection (Angeline 1998), local search (Liang & Suganthan 2005; Yuan & Zhao 2007), mutation (Andrews 2006; Alba \textit{et al.} 2007; Xue \textit{et al.} 2014), perturbation (Yuan & Zhao 2007) in BPSO to avoid pre-mature convergence. These operators are either introduced in every \textit{run} (Angeline 1998; Andrews 2006) or after a preset number of \textit{runs} (Liang & Suganthan 2005). These methods have improved the particle’s searching ability in finding an optimal solution. Nevertheless, it can be further enhanced when these methods are applied only when the particles stagnates in local optima rather than in every \textit{run} or after a fixed number of \textit{runs}.

Chuang \textit{et al.} (2008) proposed a wrapper based attribute selection technique with PSO and SVM that resets the \textit{Swarmbest} when it stagnates in local optima for a pre-fixed number of \textit{runs}. On stagnation, the authors reset \textit{Swarmbest} such that \textit{FoM} is zero (no attributes selected).
On *Swarmbest* stagnation, Vieira *et al.* (2013) reset the *Swarmbest* except at random one bit position and displaced *Ownbest* values using a parameter, \(d_r\), describing the displacement rate of *Ownbest* solution vector (i.e., probability of each bit in *Ownbest* being flipped). Though the author claims improvement in results, \(d_r\) should be well adjusted to achieve better selection.

### 2.3.2 Velocity Stagnation

When velocity of particle reaches ± \(v_{\text{max}}\), velocity stagnation occurs and the particles stagnate in a sub-optimal point in the problem space resulting in pre-mature convergence. Also, when *Swarmbest* is trapped in local optima, particles at different points in the problem space converge to local optima with large velocity as farther the target, larger the step size required to reach it. Velocity may increase to \(v_{\text{max}}\) and particles need to be diverted from local optimum which is not possible with small change in velocity. Thus, velocity stagnation becomes pre-dominant on *Swarmbest* stagnation.

Lee *et al.* (2008) and Zhang *et al.* (2014) introduced mutation operator, \(r_{\text{mut}}\), to bring random mistakes in position vector of particles similar to mutation in GA to handle the problem of velocity stagnation. This technique is effective in searching a potential solution than conventional BPSO. But, \(r_{\text{mut}}\) is fixed to \(1/D\), where, \(D\) is the attribute size and is insufficient to bring random changes in case of higher dimensionality attribute space.
2.4 GREEDY RESET AND LOCALIZED RANDOM MISTAKES

To handle the problem of pre-mature convergence in BPSO due to *Swarmbest* and velocity stagnation, BPSO with Greedy reset and localized random mistakes is proposed.

2.4.1 BPSO with Greedy Reset

BPSO with greedy *Swarmbest* reset is devised to handle the problem of pre-mature convergence to a sub-optimal solution in the problem space. It will be more meaningful to reset the *Swarmbest* based on a heuristic approach rather than resetting the *Swarmbest* with zero attributes (Chuang *et al.* 2008) or resetting all the attributes except at one random position (Vieira *et al.* 2013).

An attribute selection technique aims to find a subset with fewer attributes of high predictive capability. Of the two objectives (attribute subset size and predictive accuracy) of attribute selection, it is difficult to anticipate the predictive capability of a set of attributes without prior knowledge about the problem. It is problem specific depending on how well attributes are extracted from the problem space. Hence, it is difficult to derive a heuristic approach based on the predictive performance of attribute space to reset the *Swarmbest*. On the other hand, attribute size, $D$, of a problem is known apriori and the dimension of selected attribute subset is $D'$ with $D' < D$. On *Swarmbest* stagnation, this heuristic is adopted to design the proposed methodology to greedily look for subset with reduced attributes. On pre-mature convergence i.e., if *Swarmbest* position doesn’t change for a prefixed number of runs, *Swarmbest* is reset. It is reset such that dimension of *Swarmbest*, $k$, is reduced to $k'$ ($k' < k$). Cardinality of *Swarmbest* after reset on pre-mature convergence is one less than the *Swarmbest* before reset as in
Equation (2.8). Pseudo-code of greedy *Swarmbest* reset and *Ownbest* diversion is given in Figure 2.3.

\[ k' = k - 1 \]  
\hspace{1cm} (2.8)

**Algorithm Parameters**

- `swarm_{stag}`, Maximum allowable interval for *Swarmbest* stagnation
- `N`, Swarm size= Number of particles
- `D`, Dimension of the particle=Number of attributes

begin

if (*Swarmbest* constant for `swarm_{stag}` times)

\( k = \text{length (Swarmbest)} \)

Reset *Swarmbest*

Randomly choose \((k-1)\) positions in range \([1, D]\)

Set the chosen positions to one in the *Swarmbest*

/* *Ownbest diversion*/

for \(i = 1\) to \(N\)

\(\text{check}_i = \text{generate a number in the range } [1, D]\)

Randomly choose \(\text{check}_i\) positions in *Ownbest*$_i$

Flip only those positions chosen in the *Ownbest*$_i$

end for

end if

end

Figure 2.3 Greedy Reset and *Ownbest* Diversion
The proposed methodology resets *Swarmbest* except at \((k-1)\) positions where \(k\) is the attribute size of *Swarmbest*. Positions in *Swarmbest* for greedy reset are randomly chosen, thus holding up the stochasticity of BPSO algorithm. On reset, particles in the swarm that is bribed towards *Swarmbest* are diverted to untried areas by flipping randomly chosen positions in *Ownbest*.

### 2.4.2 BPSO with Velocity Clamping and Localized Random Mistakes

After velocity updation in BPSO, if the updated velocity is greater (lesser) than maximum (minimum) velocity, then updated velocity has to be clamped to maximum (minimum) velocity. After velocity updation, if \(v_{ij}^{run+1}\) is greater (lesser) than \(v_{max}(-v_{max})\), it is clamped to \(v_{max}(-v_{max})\). The process of velocity clamping is explained in Equation (2.9).

\[
v_{ij}^{run+1} = \max(\min(v_{max}, v_{ij}^{run+1}), -v_{max})
\]  

(2.9)

where \(v_{ij}^{run+1}\) is the updated velocity of \(i^{th}\) particle in \(j^{th}\) dimension. \(\pm v_{max}\) is the allowable step size to comb the problem space.

On velocity stagnation, position vector of particles remain unchanged and no further exploration takes place in further *runs*. Hence, localized random mistakes are introduced such that particles are diverted to unexplored areas in the problem space. Originality of solution vectors obtained on social interaction by particles in the swarm is lost if mutated in every *run*. Thus, localized random mistakes are introduced in the position vector of particles only on velocity stagnation. Pseudocode for velocity clamping and localized random mistakes is given in Figure 2.4. Number of velocity clamping per particle is calculated as ‘clamp’. From the solution vector, ‘clamp’ positions are chosen randomly and flipped, thereby, introducing localized random mistakes only during velocity stagnation.
Algorithm Parameters

\(v_{max}\), maximum allowable step size to comb the problem space

\begin{algorithm}
begin
for each particle
clamp=0
for each dimension of the particle
Update velocity, \(v_{ij}^{run+1}\) using Equation (2.3)
if \(|v_{ij}^{run+1}| \geq v_{max}\)
\(clamp = clamp + 1\)
Velocity clamping as in Equation (2.9)
end if
Update position, \(x_{ij}^{run+1}\) using Equation (2.7)
endfor
Choose ‘\(clamp\)’ number of positions in \(x_{ij}^{run+1}\) randomly
Flip only those positions chosen in \(x_{ij}^{run+1}\)
endfor
end
\end{algorithm}

Figure 2.4 Velocity Clamping and Localized Random Mistakes

Algorithm parameters for BPSO with greedy reset and localized random mistakes are given in Figure 2.5 and the overall pseudo-code is explained in Figure 2.6.
\[ N, \text{ Swarm size} = \text{Number of particles} \]
\[ D, \text{ Dimension} = \text{Number of attributes} \]
\[ C_1 \] and \[ C_2, \text{ Acceleration constants} \]
\[ v_{\text{max}}, \text{ Maximum allowable step size to comb the problem space} \]
\[ \text{swarm}_{\text{stag}}, \text{ Maximum allowable interval for Swarm best stagnation} \]
\[ \text{run}_{\text{max}}, \text{ Maximum number of runs} \]

**Figure 2.5** Algorithm Parameters for BPSO with Greedy Reset and Localized Random Mistakes

begin

Define a combinatorial problem space, \( R^D \)

Let all possible combination of attributes represent elements in \( R^D \)

Initialize algorithm parameters as in Figure 2.5

**Random initialization** of particles in the swarm, \( x_{ij}^{\text{run}} \)

Initialize \( v_{ij}^{\text{run}} \) of each particle using (2.2) and evaluate \( \text{FoM}, f_{i}^{\text{run}} \) using (1.5)

Assign fitness of each particle and its position as \( f_{\text{ownbest},i} \) and \( \text{Ownbest}_i \)

Assign best fit among particles and its position as \( f_{\text{Swarm best}} \) and \( \text{Swarm best} \)

while \( \text{run}_{\text{max}} \) is not met do

\begin{align*}
\text{for } i &= 1 \text{ to } N \\
\text{for } j &= 1 \text{ to } D \\
\end{align*}

**Figure 2.6** (Continued)
Randomly choose $r_{ij}^{run}$, $r_{2j}^{run}$, $u_{ij}^{run}$ and update velocity, $v_{ij}^{run+1}$ using (2.3)

$$\text{if } |v_{ij}^{run+1}| \geq v_{\text{max}}$$

**Velocity clamping** using Equation (2.9)

endif

Update position $x_{ij}^{run+1}$ using Equation (2.7)

end for

**Localized random mistakes** to $x_{ij}^{run+1}$

Evaluate $FoM, f_i^{run+1}$ using Equation (1.5)

if ($f_i^{run+1} \geq f_{\text{ownbest},i}$)
  $$f_{\text{ownbest},i} = f_i^{run+1} ; \text{Ownbest}_i = x_{ij}^{run+1}$$
end if

if ($f_i^{run+1} \geq f_{\text{Swarmbest}}$)
  $$f_{\text{Swarmbest}} = f_i^{run+1} ; \text{Swarmbest} = x_{ij}^{run+1}$$
end if

done for

if $\text{Swarmbest}$ constant for $\text{swarm}_{\text{stag}}$ runs
  **Greedy Swarmsbest reset** and **Ownbest Diversion**
end if

done while

done

---

**Figure 2.6 BPSO with Greedy Reset and Localized Random Mistakes**

Proposed re-initialization mechanism (Greedy reset) is significant in handling the problem of pre-mature convergence in BPSO on making a meaningful reset with dimensionality reduction as the heuristic, unlike a reset
with zero attributes (Chuang et al. 2008), a reset except at one random position (Viera et al. 2013), perturbation-a jump to a random point with random velocity (Zhan et al. 2009).

Significance of BPSO with localized random mistakes is that position vector of particles is altered only when velocity clamping occurs; unlike Andrews (2006) and Vieira et al. (2013) that introduces random mistakes invariably to all updated position vectors of particles in the swarm in every run.

The novelty of proposed methodology with Greedy reset is that it incorporates domain-specific knowledge (reduced attribute size) to reset the locked swarm best solution. Introducing greedy reset with localized random mistakes on pre-mature convergence, stochastic behaviour of swarm optimization is not lost.

The main advantage of the proposed methodology with Greedy reset and localized random mistakes is that it does not require additional parameters such as $r_{mut}$ (to introduce mutation) as in Andrews (2006) and Vieira et al. (2013) and $d_r$ (to displace Ownbest) as in Vieira et al. (2013).

2.5 EXPERIMENTATION AND RESULTS

BPSO with Greedy reset and localized random mistakes given in Figure 2.6 is implemented in Matlab 2013a and simulated on Dell Laptop with Intel i3 core processor. To investigate the success of proposed algorithm for attribute subset selection, datasets from UCI repository as tabulated in Table 1.1 are employed.
2.5.1 Data Normalization

Attributes of different dynamic ranges in a dataset are normalized so that attributes with larger values does not influence the attributes with smaller values, thus, making the attributes lie in a specified range. It also reduces difficulties during numerical calculation (Hsu et al. 2003). Min-Max normalization of data given in Equation (2.10) is used in this work to linearly scale in the range (0,1).

\[ z' = \frac{z - \text{min}_n}{\text{max}_n - \text{min}_n} \quad |z| = n \]  

(2.10)

\( z' \) is the normalized value of attribute \( z \) in the classification problem (dataset). Each attribute in a given classification problem has ‘\( n \)’ instances. \( \text{min}_n \) and \( \text{max}_n \) represent minimum and maximum values of attribute with ‘\( n \)’ instances each in the dataset.

2.5.2 Data Encoding

Presence and absence of attributes in the subset is represented by 1’s and 0’s to address the combinatorial optimization problem of attribute selection. Binary encoding of attribute space, \( x = \{x_1, x_2, x_3, x_4, x_5\} \in \mathbb{R}^5 \) with cardinality, \( |x| = 5 \), is illustrated in Figure 2.7.

![Figure 2.7 Binary Encoding of a Subset with 5 Attributes](image-url)
As observed from Figure 2.7, attributes $x_1$ and $x_3$ are selected and $x_2$, $x_4$, and $x_5$ are unselected for further processing. Attribute subset $x' = \{x_1, x_3\}$ with cardinality $|x'| = 2$.

2.5.3 Evaluation of BPSO with Greedy Reset as the Search Technique in Wrapper based Attribute Selection

The proposed algorithm for attribute selection is evaluated using datasets listed in Table 1.1 with the following parameters set in Figure 2.5.

Maximum number of runs, $\text{run}_{\text{max}}$ is set to 100 (Vieira et al. 2013). Both the acceleration constants, $C_1$ and $C_2$ are fixed to 2 (Talukder 2011) and allowable Swarmpbest stagnation before greedy reset, $\text{swarm}_{\text{stag}}$ is set to 3 (Vieira et al. 2013). Inertia weight, $\omega$ is set to 1. Swarm size, $N$ is set to 60 (empirical studies from most of BPSO implementations that use swarm size $\in [20, 60]$ Talukder 2011) and follows random initialization.

$[-v_{\text{max}}, v_{\text{max}}]$ is set to $[-6, 6]$ such that sigmoid transformation in Equation (2.6) yields a probability range $[0.9975, 0.0025]$. Thus, velocity $v_{ij}^{\text{run}}$ is the probability of the particle’s position, $x_{ij}^{\text{run}}$ to remain at 0 or 1 (Kennedy & Eberhart 1997).

SVM with radial basis kernel (Hsu et al. 2003) is used for subset evaluation with 10 fold cross-validation. Predictive accuracy and the corresponding subset size of Musk dataset is shown in Figure 2.8a and Figure 2.8b respectively for single iteration with 100 runs.
Figure 2.8  Predictive Accuracy (a) and Size of Attribute Subset (b) that are realized in 100 runs for Musk Dataset on applying BPSO with Greedy Reset and Localized Random Mistakes

The following observations are made from Figure 2.8:

i) As evident from Figure 2.8a, on pre-mature convergence, *Swarmbest* is reset to another point (new *Swarmbest*) in the problem space. From Figure 2.8b, it is observed that the new *Swarmbest* selected has one attribute less than the past *Swarmbest*, resulting in greedy reset and search.

ii) It is also observed from Figure 2.8b, that the proposed algorithm, in the course of the search, greedily looks for subsets with lesser attributes than the previous *Swarmbest* in the problem space (from run 1-27). At the same time, greedy reset and search do not compromise on predictive accuracy as evident at run 28 because *FoM* calculated using Equation (1.5) is modelled with more importance laid to accuracy than attribute subset size.
2.5.4 Accuracy Gain and Dimensionality Reduction by Wrapper Approach using BPSO with Greedy Reset

Predictive accuracy of SVM classifier without dimensionality reduction (all the attributes as input) is calculated to compare the predictive accuracy of the classifier after dimensionality reduction. The results are tabulated across Table 2.1 in the form of average ± standard deviation. Predictive accuracy of selected attributes, from the process of attribute selection is compared with predictive accuracy of SVM classifier with all the attributes as input to calculate Accuracy Gain (AG) in employing wrapper based attribute selection technique. It is calculated as the improvement in predictive accuracy with selected attributes than with all the attributes in the dataset.

\[
\text{Accuracy Gain(\%)} = \frac{P_{\text{sel}} - P}{P_{\text{sel}}} \times 100
\]  
(2.11)

where \(P_{\text{sel}}\) and \(P\) represent predictive accuracy of SVM classifier with selected attributes and all the attributes as input.

Similarly, Dimensionality Reduction (DR) is calculated as the reduction in dimensionality of attribute space on employing wrapper based attribute selection technique.

\[
\text{Dimensionality Reduction} = \frac{\#A - \#A_{\text{sel}}}{\#A} \times 100
\]  
(2.12)

where \(\#A_{\text{sel}}\) and \(\#A\) represent number of selected attributes and total number of attributes in the original dataset.

Of the 12 datasets, 6 datasets (Australian, WBCD, Spectf, Sonar, Hill Valley and Musk) shows an accuracy gain of more than 5% and only 3 datasets (Hepatitis, WBC and Spectf) shows dimensionality reduction less
than 75%. There is a commendable reduction in dimensionality of attribute space for decision making in the final classification phase.

Table 2.1  Accuracy Gain (AG) and Dimensionality Reduction (DR) by Wrapper Approach using BPSO with Greedy Reset

<table>
<thead>
<tr>
<th>Data</th>
<th>Without Attribute Selection (Average±Standard Deviation)</th>
<th>Proposed (Average±Standard Deviation)</th>
<th>AG (%)</th>
<th>DR (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P (%) #A</td>
<td>P (%) #A</td>
<td>AG</td>
<td>DR</td>
</tr>
<tr>
<td>PID</td>
<td>72.25±0.72 8</td>
<td>74.03±0.3 1.5±0.51</td>
<td>2.4</td>
<td>81.25</td>
</tr>
<tr>
<td>WBCO</td>
<td>95.39±0.64 9</td>
<td>95.95±0.32 2.2±0.57</td>
<td>0.58</td>
<td>75.55</td>
</tr>
<tr>
<td>Australian</td>
<td>82.55±0.94 14</td>
<td>87.53±0.71 3.2±0.3</td>
<td>5.68</td>
<td>77.14</td>
</tr>
<tr>
<td>Hepatitis</td>
<td>74.64±0.84 19</td>
<td>74.75±0.72 7.8±0.6</td>
<td>0.14</td>
<td>58.94</td>
</tr>
<tr>
<td>Credit</td>
<td>69.88±0.81 24</td>
<td>70.50±0.5 5.5±0.2</td>
<td>0.87</td>
<td>77.08</td>
</tr>
<tr>
<td>WBCD</td>
<td>90.06±0.32 30</td>
<td>94.81±0.81 2.8±0.7</td>
<td>5.01</td>
<td>90.67</td>
</tr>
<tr>
<td>WBC</td>
<td>76.69±0.71 32</td>
<td>76.85±0.91 8.5±0.5</td>
<td>0.20</td>
<td>73.43</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>87.57±0.98 34</td>
<td>92.14±0.46 7.2±0.92</td>
<td>4.95</td>
<td>78.82</td>
</tr>
<tr>
<td>Spectf</td>
<td>78.95±0.94 44</td>
<td>83.27±0.72 11.5±1.39</td>
<td>5.18</td>
<td>73.86</td>
</tr>
<tr>
<td>Sonar</td>
<td>56.60±0.74 60</td>
<td>86.77±0.92 10.3±1.2</td>
<td>34.77</td>
<td>82.83</td>
</tr>
<tr>
<td>Hill Valley</td>
<td>46.76±0.62 100</td>
<td>52.06±0.72 1.6±0.83</td>
<td>10.18</td>
<td>98.40</td>
</tr>
<tr>
<td>Musk</td>
<td>56.99±0.32 166</td>
<td>86.92±0.37 18.1±3.54</td>
<td>34.43</td>
<td>89.06</td>
</tr>
</tbody>
</table>

Datasets with more than 50 attributes (Spectf, Hill Valley and Musk) achieves an accuracy gain of more than 10% and dimensionality reduction of more than 82%. Thus, the proposed attribute selection methodology favours large dimensional datasets with superior performance in terms of both accuracy gain and dimensionality reduction. Although datasets
such as WBCO, Hepatitis, Credit and WBC show very meagre improvement in accuracy, there is remarkable reduction in dimensionality of attribute space.

2.5.5 BPSO with Greedy Reset Vs Literature on BPSO for Attribute Selection

The performance of proposed algorithm is compared with literature on BPSO for attribute subset selection. Results of existing and proposed work using BPSO are tabulated in Table 2.2.

Table 2.2 BPSO with Greedy Reset Vs Literature on BPSO for Attribute Selection

<table>
<thead>
<tr>
<th>Data / #A</th>
<th>Proposed (Average±Standard Deviation)</th>
<th>Literature</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PID /8</td>
<td>P (%) ±0.3 1.5±0.51 74.03±0.3 1.5±0.51</td>
<td>71.8 4</td>
<td>Unler &amp; Murat (2010)</td>
</tr>
<tr>
<td>WBCO /9</td>
<td>P (%) ±0.32 2.2±0.57 95.95±0.32 2.2±0.57</td>
<td>96.31 2</td>
<td>Lee et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.54 2</td>
<td>Chuang et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87 4</td>
<td>Unler &amp; Murat (2010)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.34 2</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.34 2.79</td>
<td>Vieira et al. (2013)</td>
</tr>
<tr>
<td>Australian /14</td>
<td>P (%) ±0.71 3.2±0.3 87.53±0.71 3.2±0.3</td>
<td>84.24 3.42</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td>Hepatitis /19</td>
<td>P (%) ±0.72 7.8±0.6 74.75±0.72 7.8±0.6</td>
<td>89.7 13</td>
<td>Yun et al. (2011)</td>
</tr>
<tr>
<td>Credit/24</td>
<td>P (%) ±0.5 5.5±0.2 70.50±0.5 5.5±0.2</td>
<td>69.15 11.92</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75.6 22.4</td>
<td>Lin &amp; Chen (2009)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>69.15 11.92</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>76 3</td>
<td>Vieira et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>72.07 4</td>
<td>Lee et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>74.61 3</td>
<td>Chuang et al. (2008)</td>
</tr>
<tr>
<td>Data / #A</td>
<td>Proposed (Average±Standard Deviation)</td>
<td>Literature</td>
<td>Reference</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------</td>
<td>------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td>P (%) #A</td>
<td>P (%) #A</td>
<td></td>
</tr>
<tr>
<td>WBCD/30</td>
<td>94.81±0.81 2.8±0.7</td>
<td>98.36 12.9</td>
<td>Zhang et al. (2015)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.66 12.2</td>
<td>Unler et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>93.98 3.46</td>
<td>Xue et al. (2014)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>98.17 14.2</td>
<td>Chuang et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.8 4</td>
<td>Unler &amp; Murat (2010)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>95.61 13</td>
<td>Tu et al. (2007)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>93.54 5</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.71 3</td>
<td>Vieira et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.85 3</td>
<td>Lee et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.34 3</td>
<td>Chuang et al. (2008)</td>
</tr>
<tr>
<td>WBC/32</td>
<td>76.85±0.91 8.5±0.5</td>
<td>78.9 3</td>
<td>Vieira et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.53 3</td>
<td>Lee et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.04 3</td>
<td>Chuang et al. (2008)</td>
</tr>
<tr>
<td>Ionosphere/34</td>
<td>92.14±0.46 7.2±0.92</td>
<td>92.3 17</td>
<td>Yun et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.21 11.4</td>
<td>Zhang et al. (2015)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>95.54 9.25</td>
<td>Unler et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87.27 3.26</td>
<td>Xue et al. (2014)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>92.2 21.7</td>
<td>Lin &amp; Chen (2009)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87.52 13.79</td>
<td>Esseghir et al. (2010)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>97.33 15</td>
<td>Tu et al. (2007)</td>
</tr>
<tr>
<td>Spectf/44</td>
<td>83.27±0.72 11.5±1.39</td>
<td>79.4 8</td>
<td>Yun et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>83.46 15</td>
<td>Inbarani et al. (2014)</td>
</tr>
<tr>
<td>Data / #A</td>
<td>Proposed (Average±Standard Deviation)</td>
<td>Literature</td>
<td>Reference</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------</td>
<td>------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td>P (%) #A</td>
<td>P (%) #A</td>
<td></td>
</tr>
<tr>
<td><strong>Sonar /60</strong></td>
<td>86.77±0.92 10.3±1.2</td>
<td>96.08 28.2</td>
<td>Zhang et al. (2015)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.67 13.9</td>
<td>Unler et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>78.16 11.24</td>
<td>Xue et al. (2014)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>90.5 38.1</td>
<td>Lin &amp; Chen (2009)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.92 30.2</td>
<td>Chuang et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>73.59 20.73</td>
<td>Esseghir et al. (2010)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>96.25 34</td>
<td>Tu et al. (2007)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>93.74 12</td>
<td>Vieira et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>88.86 13</td>
<td>Lee et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>91.79 12</td>
<td>Chuang et al. (2008)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>84.6 28</td>
<td>Yun et al. (2011)</td>
</tr>
<tr>
<td><strong>Hill Valley /100</strong></td>
<td>52.06±0.72 1.6±0.83</td>
<td>57.77 12.22</td>
<td>Xue et al. (2014)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>57.57 47.05</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td><strong>Musk /166</strong></td>
<td>86.92±0.37 18.1±3.54</td>
<td>84.58 85.58</td>
<td>Xue et al. (2013)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>85.9 79</td>
<td>Yun et al. (2011)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>84.94 76.54</td>
<td>Xue et al. (2014)</td>
</tr>
</tbody>
</table>

Proposed attribute selection algorithm shows superior results over most of the literature on BPSO for the following reasons:

i) At every *Swarmbest* stagnation, particles are reset so as to find a subset of attributes one less than the past, with a motive of dimensionality reduction.

ii) Originality of position updation obtained on efficient exploration by particles in the swarm is unaltered, unless there is velocity stagnation.
2.5.6 Performance of Various Classifiers with Selected Attributes by BPSO with Greedy Reset

The predictive capability of 6 traditional classifiers with selected attributes as input is measured to validate the efficiency of the proposed attribute selection technique as a pre-processing tool to classification. Attribute subset selected by the proposed algorithm is input to 6 traditional classifiers such as SVM (RBF, \( \sigma = 1 \)) with \( c = 1 \), \( kNN \) with \( k=1 \), NB, DA, DT and Ensemble learning with Ada Boost algorithm. Ten fold cross-validation is employed and the results are averaged over 100 iterations. Results are consolidated in Table 2.3.

Table 2.3 Predictive Performance of 6 Classifiers with Attributes Selected by BPSO with Greedy Reset

<table>
<thead>
<tr>
<th>Data Label</th>
<th>SVM (%)</th>
<th>KNN (%)</th>
<th>DA (%)</th>
<th>NB (%)</th>
<th>DT (%)</th>
<th>Ensemble (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_1</td>
<td>74.2</td>
<td>68.2</td>
<td>73.0</td>
<td>74.1</td>
<td>69.1</td>
<td>73.8</td>
</tr>
<tr>
<td>D_2</td>
<td>96.2</td>
<td>95.1</td>
<td>95.4</td>
<td>95.8</td>
<td>95.4</td>
<td>96.1</td>
</tr>
<tr>
<td>D_3</td>
<td>87.2</td>
<td>82.1</td>
<td>86.1</td>
<td>86.5</td>
<td>83.2</td>
<td>86.9</td>
</tr>
<tr>
<td>D_4</td>
<td>75.0</td>
<td>72.3</td>
<td>74.8</td>
<td>71.0</td>
<td>75.2</td>
<td>75.4</td>
</tr>
<tr>
<td>D_5</td>
<td>71.1</td>
<td>69.1</td>
<td>70.8</td>
<td>70.2</td>
<td>69.4</td>
<td>70.1</td>
</tr>
<tr>
<td>D_6</td>
<td>94.7</td>
<td>92.9</td>
<td>94.8</td>
<td>93.2</td>
<td>94.4</td>
<td>94.2</td>
</tr>
<tr>
<td>D_7</td>
<td>77.2</td>
<td>75.0</td>
<td>75.2</td>
<td>75.4</td>
<td>71.2</td>
<td>75.8</td>
</tr>
<tr>
<td>D_8</td>
<td>93.2</td>
<td>92.2</td>
<td>89.2</td>
<td>89.1</td>
<td>92.4</td>
<td>92.5</td>
</tr>
<tr>
<td>D_9</td>
<td>83.1</td>
<td>80.1</td>
<td>79.2</td>
<td>80.9</td>
<td>81.2</td>
<td>83.3</td>
</tr>
<tr>
<td>D_{10}</td>
<td>86.7</td>
<td>84.2</td>
<td>83.8</td>
<td>84.3</td>
<td>84.8</td>
<td>85.1</td>
</tr>
<tr>
<td>D_{11}</td>
<td>52.1</td>
<td>51.5</td>
<td>51.0</td>
<td>49.1</td>
<td>51.6</td>
<td>52.1</td>
</tr>
<tr>
<td>D_{12}</td>
<td>86.1</td>
<td>82.7</td>
<td>83.2</td>
<td>81.8</td>
<td>84.3</td>
<td>85.0</td>
</tr>
</tbody>
</table>
All the six classifiers achieve good predictive accuracy with selected attributes as input. SVM and Ensemble learning classifiers show better results than other classifiers owing to their better generalization and predictive ability. Selected attributes, when fed as input to traditional classifiers (kNN, NB, DT, DA and Ensemble) yield results in par with that achieved by attribute selection technique employing SVM classifier. This proves the generality of the proposed algorithm as a pre-processing tool before any learning algorithm.

## 2.6 SUMMARY

A wrapper based attribute selection methodology using BPSO and SVM is implemented to select highly predictive attributes as input for the final classification phase in any pattern recognition problem. BPSO employed for optimal attribute subset search, suffers from partial optimization. A novel re-initialization technique named Greedy reset is proposed to avoid premature convergence to a sub-optimal solution during the search. Proposed methodology explores the problem space for an optimal attribute subset without employing additional parameters.

The efficiency of proposed attribute selection algorithm is tested and validated over 12 benchmark datasets from UCI repository representing classification problems. To prove the efficiency of proposed algorithm, performance indices namely accuracy gain and dimensionality reduction are calculated on comparing the predictive capability of attributes before and after attribute selection.

The performance of proposed algorithm is compared with literature on BPSO for attribute selection. Simulation results show that, in most of the instances, the proposed algorithm performs better compared to existing attribute selection algorithm using BPSO. Finally, predictive capability of 6
traditional classifiers with selected attributes are measured. It is observed that in par results are achieved by all the classifiers, thus proving the generality of the proposed algorithm.