Chapter 6

Butterfly Optimizer

6.1 Introduction

In this chapter, a recent swarm optimization method known as Butterfly Optimizer(BO) [81, 82], has been applied for simultaneous feature selection to get optimal reduct. This algorithm secured first position at a special session and competition on single objective bound constraint optimization organized in IEEE CEC 2017. This algorithm performs better than other popular variants of differential evolution and covariance matrix adaptation evolution strategies (well developed and popular meta-heuristics). This was the reason behind the consideration of this algorithm as a feature selection algorithm.

This feature selection method uses a population of reducts. For each of the candidate reduct, fitness function is evaluated as a function of (i) fuzzy rough dependency measure, $\gamma'_P(Q)$, and (ii) cardinality, $|R|$, of the reduct R (Equation 4.5). Fuzzy rough dependency measure, $\gamma'_P(Q)$, for reduct P corresponding to class label, Q, is calculated using method of rough and fuzzy lower approximation based fuzzy rough set (Equations 3.9 and 3.21). In this work, a reduct is represented as a binary string, for doing the feature selection using BO.

Butterfly Optimizer (BO) [81] is a dual population based technique for unconstrained optimization. BO is based on dual population of positions of male butterflies, that undergo operations of *perching* and *patrolling*. Perching and patrolling operations of BO correspond to exploration and exploitation of search space respectively, to look for a new solution. In this string, each digit "1" corresponds to the selected feature.

6.2 Butterfly Optimizer

The brief discussion of the BO [81] technique is given as follows.

Dual population of BO

BO is based on a dual population of positions of male butterflies. Perching and patrolling operations of BO correspond to exploration and exploitation of search space respectively, to look for new solution. For $D-$ dimensional problem, with N butterflies, population P_1 represents current perching positions and, population P_2 consists of best perching positions achieved so far of every male butterfly. P_1 and P_2 are represented as $N \times D$ matrix as follows.

$$P_1^k = [\bar{x}_1^k, \bar{x}_2^k....\bar{x}_N^k] \tag{6.1}$$

where

$$\bar{x}_i^k = [x_{i1}^k, x_{i2}^k, x_{i3}^k....x_{iD}^k]^T, i = 1, 2, ...N \tag{6.2}$$

and

$$P_2^k = [m\bar{x}_1^k, m\bar{x}_2^k....m\bar{x}_N^k] \tag{6.3}$$

where

$$m\bar{x}_i^k = [mx_{i1}^k, mx_{i2}^k, mx_{i3}^k....mx_{iD}^k]^T, i = 1, 2, ...N \tag{6.4}$$

The vector $\bar{x}_i^k$ and $m\bar{x}_i^k$, and thereby P_1 and P_2 are updated using BO algorithm to get newer solutions.

Initialization

Solution process is initialized in search space using Equation (6.5).

$$x_{ij}^0 = (U_j - L_j) \times rand + L_j \tag{6.5}$$

$$mx_{ij}^0 = (U_j - L_j) \times rand + L_j$$

$$mv_{ij}^0 = 0$$

For j^{th} parameter, L_j and U_j are lower and upper bound and $rand$ is uniformly distributed random number between $(0, 1]$

Perching

Perching operation consists of (i) *Crisscross modification* and (ii) *I-selection.*

Crisscross Modification:

Crisscross modification updates the perching position vector $\bar{x}_i^{k+1}$, by modifying one of its randomly selected elements, d, in the following manner.

$$x_{ij}^{k+1} = \begin{cases} \Re(x_{cc_ij}^k, mx_{cc_ij}^k) + F * (\Re(x_{q_ij}^k, mx_{q_ij}^k) - \Re(x_{r_ij}^k, mx_{r_ij}^k)), & \text{if } j == d \\ mx_{ij}^k, & \text{otherwise.} \end{cases} \tag{6.6}$$

$\Re(*, *)$, is a random operator, which can pick one of the arguments with equal probability. Crisscross neighbor of i^{th} butterfly is cc_i. Other randomly selected neighbors of i^{th} butterfly, q_i and r_i satisfy the following criteria.

$$i \neq cc_i \neq q_i \neq r_i \tag{6.7}$$

At the beginning of an iteration, cc_i of length N is initialized by randomly shuffling integers from 1 to N.

$$\bar{cc} = [cc_1, cc_2, cc_3....cc_i]^T, i = 1, 2, 3...N \tag{6.8}$$

I-Selection:

I-selection is given by Equation (6.9) which updates $\bar{mx}_i^k$ of P_2 as follows.

$$\bar{mx}_i^{k+1} = \begin{cases} \bar{x_i}^{k+1}, & \text{if } f(\bar{x_i}^{k+1}) \leq f(\bar{mx_i}^k) \\ \bar{mx_i}^k, & \text{otherwise.} \end{cases} \tag{6.9}$$

where $f(*)$ is the objective function value at $*$ position.

Patrolling

Male butterflies which remain *un*-updated during perching operation are updated in patrolling operation. Patrolling operation consists of following two sub-operations. *Towards-best modification:*

This step gives patrolling position vector, $\bar{ux}_i$

$$\bar{ux}_i = \bar{mx}_i^k + s * \bar{mv}_i^k + F * (\bar{mx}_{maxuv_i}^k - \bar{mx}_i^k), \tag{6.10}$$

where $\bar{mx}_{maxuv_i}$ is the best position in the population, F is a random value between $(0, 1]$ and s is a constant between $[0, 1]$.

II-Selection:

II-Selection is similar to I-selection; the only difference is that it also updates a velocity vector $\bar{mv}_i^k$. Equation (6.11) updates velocity vector as follows,

$$\bar{mv}_i^{k+1} = \begin{cases} \bar{ux}_i^{k+1} - \bar{mx}_i^k, & \text{if} f(\bar{ux_i}^{k+1}) \leq f(\bar{mx_i}^k) \\ dd * \bar{mv}_i^k + F * (\bar{mx}_{maxuv_i}^k - \bar{mx}_i^k), & \text{otherwise.} \end{cases} \tag{6.11}$$

BO algorithm is terminated if specified maximum number of objective function evaluation is reached or solution does not change over a specified number of consecutive iterations.

Algorithm

The algorithm for Butterfly Optimizer (BO) method is given in *Algorithm 10.* In between the searching process this algorithm calls *Algorithm 8* for perching and *Algorithm 9* for patrolling.

The salient similarities and differences among the methods of PSO, DE, and BO are discussed below and it has been explained that in what respect BO method is better than PSO and DE.

PSO: A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

DE: Differential Evolution (DE), proposed by Storn and Price is a stochastic population-based search method. It exhibits excellent capability in solving a wide range of optimization problems with different characteristics from several fields and many real-world application problems. Similar to all other Evolutionary algorithms (EAs), the evolutionary process of DE uses mutations, crossover and selection operators at each generation to reach the global optimum.

In DE, each individual in the population is called the target vector. Mutation is used to generate a mutant vector, which perturbs a target vector using the difference vector of other individuals in the population. After that, crossover operation generates

Algorithm 8 Perching

procedure PERCHING()

$cc_i \leftarrow randP(i)$

$q_i \leftarrow randi(1, N)$

while $(cc_i == q_i)$ or $(i == q_i)$ **do**

$q_i \leftarrow randi(1, N)$

end while

$r_i \leftarrow randi(1, N)$

$/{*}i \neq cc_i \neq q_i \neq r_i{*}/$

while $(cc_i == r_i)$ or $(q_i == r_i)$ or$(i == r_i)$ **do**

$r_i \leftarrow randi(1, N)$

end while

$d_i \leftarrow randi[1, D]$

$F \leftarrow rand(0, 1]$

for $j = 1$ to D **do**

if $(j == d)$ **then**

$x_{ij}^{k+1} = \Re(x_{cc_ij}^k, mx_{cc_ij}^k) + F * (\Re(x_{q_ij}^k, mx_{q_ij}^k) - \Re(x_{r_ij}^k, mx_{r_ij}^k))$

else

$x_{ij}^{k+1} = mx_{ij}^k;$

end if

end for

/*$\Re(*, *)$, is a random operator, which can pick one of the arguments with equal probability*/

$f(\bar{x_i}^{k+1}) \leftarrow$ function evaluation at $\bar{x_i}^{k+1}$

if $(f(\bar{x_i}^{k+1}) \leq f(m\bar{x_i}^k))$ **then**

$m\bar{x}_i^{k+1} = \bar{x_i}^{k+1}$

else

$m\bar{x}_i^{k+1} = m\bar{x_i}^k$

end if

end procedure

Algorithm 9 Patrolling

procedure PATROLLING()

  for $j = 1$ to D **do**

    $F \leftarrow$ uniformly distributed number from 0 to 1

    $\bar{ux}_i = \bar{mx}_i^k + s * \bar{mv}_i^k + F * (\bar{mx}_{maxuv_i}^k - \bar{mx}_i^k)$

  end for

  $f(\bar{ux}_i) \leftarrow$ function evaluation at $\bar{ux}_i$

  if $f(\bar{ux}_i) \leq f(\bar{mx}_i{}^k)$ **then**

    $\bar{mv}_i^{k+1} = \bar{ux}_i - \bar{mx}_i^k$

  else

    $\bar{mv}_i^{k+1} = dd * \bar{mv}_i^k + F * (\bar{mx}_{maxuv_i}^k - \bar{mx}_i^k)$

  end if

end procedure

the trial vector by combining the parameters of the mutation vector with the parameters of a parent vector selected from the population. Finally, according to the fitness value and selection operation determines which of the vectors should be chosen for the next generation by implementing a one-to-one completion between the generated trail vectors and the corresponding parent vectors. The performance of DE basically depends on the mutation strategy, the crossover operator.

BO: Butterfly Optimizer (BO) is a dual population based technique for unconstrained optimization. BO is based on the dual population of positions of male butterflies, which undergo operations of perching and patrolling. Perching and patrolling operations of BO correspond to exploration and exploitation of search space respectively, to look for a new solution.

Perching operation consists of (i) Crisscross modification and (ii) I-selection. Male butterflies which remain un-updated during perching operation are updated in patrolling operation using II-selection.

The performance of the three methods discussed above depends on the search operators designed for exploration and exploitation of search space. In PSO the operators are simple but quite crude in the sense that the operators do not take care of manipulation of the subcomponents of the particles. Any two particles interact as a whole to produce a

Algorithm 10 Butterfly Optimizer

procedure BO()

  procedure INITIALIZATION()

    for $i = 1$ to N **do**

      for $j = 1$ to D **do**

        $x_{ij}^0 = (U_j - L_j) * rand + L_j$

        $mx_{ij}^0 = (U_j - L_j) * rand + L_j$

        $mv_{ij}^0 = 0$

      end for

      $\bar{x}_i^k \leftarrow [x_{i1}^k, x_{i2}^k, x_{i3}^k....x_{iD}^k]^T$, $i = 1, 2, ...N$

      $m\bar{x}_i^k \leftarrow [mx_{i1}^k, mx_{i2}^k, mx_{i3}^k....mx_{iD}^k]^T$, $i = 1, 2, ...N$

    end for

    $P_1^k \leftarrow [\bar{x}_1^k, \bar{x}_2^k....\bar{x}_N^k]$

    $P_2^k \leftarrow [m\bar{x}_1^k, m\bar{x}_2^k....m\bar{x}_N^k]$

  end procedure

  while termination condition is not satisfied **do**

    $maxUV \leftarrow$ most attractive butterfly

    for $i = 1$ to $N/2$ **do**

      $randP \leftarrow randperm(N)$

      PERCHING()

      if position of i^{th} butterfly is not updated **then**

        PATROLLING()

      end if

    end for

    $P \leftarrow [P_1|P_2]$

  end while

end procedure

new particle. In the DE also, the operators are similar to PSO, however, the major shift takes place by the introduction of an operator which manipulates the subcomponents of each of the particles; this operator is named as mutation. In BO, a more elaborate arrangement of operators is found as compared to PSO and DE, which take care of components and subcomponents and which are based on the performance of the individual butterflies (particles) throughout generations. Thus, the working of PSO, DE, and BO can be summarized as follows. (i) PSO: particles are manipulated/updated as a whole, (ii) DE: Particles and subcomponents of the particles are manipulated/updated, and (iii) BO: Particles and subcomponents of the particles are manipulated/updated as well as particles not changing over the generations are also manipulated/updated.

Thus BO updates the solution (particles) more efficiently due to the involvement of several operators. The detailed verification of the performance of BO compared to PSO and DE has been illustrated in terms of (i) Best of Optimum Value, found after Maximum function evaluation by BO, DE, PSO, and ABC for 30 Independent Runs, for Benchmark problems, (ii) Mean of Optimum value, found after Maximum function evaluation (500000) by BO, DE, PSO, and ABC for 30 Independent Runs, for Benchmark Problems, and (iii) Standard Deviation of Optimum value, found after Maximum function evaluation (500000) by BO, DE, PSO, and ABC for 30 Independent Runs, for Benchmark Problems [81].

In this section, BO is introduced as a feature selection algorithm as the performance of BO is far better that that of DE and PSO [81, 82]. Moreover, this algorithm secured first position at special session and competition on single objective bound constraint optimization organized in IEEE CEC 2017 [82]. This algorithm perform better than other popular variants of differential evolution and covariance matrix adaptation evolution strategy (well developed and popular meta heuristics). This was the reason behind the consideration of this algorithm as a feature selection algorithm.

6.3 Results and Discussions

6.3.1 BO method with RST measure

To validate the performance of BO, the results of feature selection obtained using BO algorithm have been compared with results of feature selection using Hybrid-P and Hybrid-S discussed in the previous chapter. In this section, results for feature selection using the BO method have been presented using rough dependency measure of RST. Note that the parameters of BO are set by using sensitivity analysis, where $dd = 0.7$, s and F are randomly generated between -1 to 1. Superiority of the BO method, w.r.t Hybrid-P and Hybrid-S methods of feature selection is established through the results shown in Table 6.3 and following discussions.

Tables 6.1 and 6.2 present the results of feature selection using the BO method. The results of the feature selection are presented in Table 6.1 in the terms of number of features in the reducts suggested by each of the methods. For the 12 runs of each method for any dataset, best reduct size, mean reduct size and standard deviation (s.d.) among the reduct sizes have been presented.

In Tables 6.1 and 6.2, t-test (a parametric test) and Wilcoxon test (non-parametric test) have also represented. the significance value of 0.05 has been taken in these tests.

It is observed from Tables 6.1 and 6.2 that in the case of *Cleveland*, *Ecoli*, *Glass*, *Ionosphere*, *Soybean small* and *Wine* datasets, BO is as good as other methods, in terms of reduct size and classification accuracy. In the case of *Lung* the t-test as well as Wilcoxon test for reduct size (Table 6.1) suggest that BO performs better than Hybrid-P but have comparable performance w.r.t Hybrid-S methods. In the case of *LSVT* dataset, the BO method improves the result significantly, in terms of number of features (Table 6.1). For *Soybean-small*, BO gives guaranteed optimal result i.e. reduct size as 2. In almost all the datasets BO has better performance, especially in the cases of *Ionosphere*, *Lung*, *Soybean small*, *Wine* and *LSVT* datasets.

Tables 6.1 and 6.2 show best results reported in literature in terms of Mean Subset size (denoted as MSS) and classification accuracy (denoted as CA). Further, it is evident from the Tables 1.1, 6.1 and 6.2 that the performance of BO is better than that of state-of-the-art methods suggested in literature. While using rough dependency measure as fitness function, these method provides smaller reduct with comparable or more accuracy than the existing best method reported in literature, for all the dataset. BO produces the stable result with zero or very low standard deviation.

Table 6.3 shows the results of Friedman test giving the ranking of the performance of all the methods. It is observed that BO ranks the best as compared to Hybrid-P and Hybrid-S methods.

Table 6.1: BO with RST measure: Comparison of reduct size with statistical t-test, and Wilcoxon test. [Statistical t-test is denoted as 'S'. Wilcoxon test is denoted as 'WT'. Mean Subset Size is denoted as 'MSS'.]

Dataset (Total Features)	Feature Selection Method	Feature Subset size				Best MSS reported in literature
		Min	Mean(s.d)	S	W.T.	
Cleveland(13)	Hybrid-P	3	3(0)	-	-	7.81 [30]
	Hybrid-S	3	3(0)	-	-	
	BO	**3**	3(0)			
Ecoli(7)	Hybrid-P	3	3(0)	-	-	3 [52]
	Hybrid-S	3	3(0)	-	-	
	BO	**3**	3(0)			
Glass(9)	Hybrid-P	2	2(0)	-	-	8.44 [30]
	Hybrid-S	2	2(0)	-	-	
	BO	**2**	2(0)			
Ionospere(34)	Hybrid-P	2	2.58(0.515)	-	-	7.3 [30]
	Hybrid-S	2	2.25(0.45)	-	-	
	BO	**2**	**2(0)**			
Lung(56)	Hybrid-P	4	4.67(0.49)	v	v	NA
	Hybrid-S	3	4.58(0.668)	-	-	
	BO	**3**	**3(0)**			
Soybean small(35)	Hybrid-P	2	2(0)	-	-	2 [29]
	Hybrid-S	2	2(0)	-	-	
	BO	**2**	2(0)			
Wine(13)	Hybrid-P	2	2(0)	-	-	2 [52]
	Hybrid-S	2	2(0)	-	-	
	BO	**2**	2(0)			
LSVT(310)	Hybrid-P	3	8.58(3.11)	-	-	NA
	Hybrid-S	12	14.75(2.3)	v	v	
	BO	**4**	**8.34(2.2)**			

Table 6.2: BO with RST measure: Comparison of classification accuracy with statistical t-test, and Wilcoxon test. [Classification accuracy is denoted as 'CA'. Statistical t-test is denoted as 'S'. Wilcoxon test is denoted as 'WT'.]

Dataset (Total features)	Feature Selection Method	Classification Accuracy (CA)												Best CA reported in literature
		Classifier: J48				Classifier: JRip				Classifier: PART				
		Max	Mean(s.d)	S	W.T.	Max	Mean(s.d)	S	W.T.	Max	Mean(s.d)	S	W.T.	
Cleveland(13)	Hybrid-P	55.44	51.61(2.406)	-	-	53.46	52.66(0.432)	-	-	54.78	51.28(2.652)	-	-	52.6 [30]
	Hybrid-S	55.44	51.81(3.365)	-	-	53.46	52.99(0.432)	-	-	54.78	50.84(2.892)	-	-	
	BO	55.44	52.67(1.62)			**56.1**	52.79(1.14)			54.78	51.56(2.21)			
Ecoli(7)	Hybrid-P	79.46	76.8(3.5)	-	-	80.95	76.33(4.98)	-	-	80.95	76.8(5.2)	-	-	77.38 [52]
	Hybrid-S	79.46	75.17(4.13)	-	-	80.95	73.51(5.1)	-	-	80.95	73.86(5.65)	-	-	
	BO	79.46	77.22(2.51)			**80.95**	77.59(2.27)			80.95	77.31(1.1)			
Glass(9)	Hybrid-P	66.36	60.39(5.36)	-	-	64.95	59.38(6.3)	-	-	68.22	59.89(6.47)	-	-	65.14 [30]
	Hybrid-S	66.36	61.68(3.81)	-	-	64.95	59.15(3.87)	-	-	68.22	60.75(5.05)	-	-	
	BO	66.36	62.46(2.02)			64.95	59.17(3.01)			**68.22**	61.64(3.17)			
Ionospere(34)	Hybrid-P	**90.31**	86.2(2.36)	-	-	89.46	86.51(2.02)	-	-	89.17	85.78(2.36)	-	-	86.17 [30]
	Hybrid-S	88.89	84.95(2.16)	-	-	88.32	85.23(1.32)	-	-	87.75	83.99(2.09)	-	-	
	BO	88.89	86.71(2.31)			88.89	86.53(1.63)			87.75	85.85(2.9)			
Lung(56)	Hybrid-P	84.37	71.35(8.085)	v	v	84.37	72.65(5.976)	v	v	81.25	70.31(6.321)	v	v	NA
	Hybrid-S	87.5	76.82(10.86)	-	-	87.5	75.25(9.37)	-	-	87.5	76.3(8.37)	-	-	
	BO	**87.5**	86.27(2.11)			87.5	85.82(2.16)			87.5	83.9(2.32)			
Soybean small(35)	Hybrid-P	100	100(0)	-	-	100	100(0)	-	-	100	100(0)	-	-	100 [29]
	Hybrid-S	100	100(0)	-	-	100	100(0)	-	-	100	100(0)	-	-	
	BO	**100**	100(0)			100	100(0)			100	100(0)			
Wine(13)	Hybrid-P	84.83	81.6(2.85)	-	-	83.71	80.95(3.02)	-	-	85.39	80.95(3.58)	-	-	90.44 [52]
	Hybrid-S	93.82	79.78(8.36)	-	-	90.45	77.8(7.003)	-	-	93.82	79.59(8.64)	-	-	
	BO	**94.94**	85.39(1.31)			90.45	86.37(2.45)			93.82	85.45(1.73)			
LSVT(310)	Hybrid-P	**84.92**	76.19(4.59)	-	-	84.92	76.52(4.51)	-	-	80.95	76.32(4.04)	-	-	NA
	Hybrid-S	84.13	75.46(3.67)	-	-	80.16	74.47(3.46)	-	-	80.16	74.14(3.75)	-	-	
	BO	80.16	77.12(2.12)			80.95	77.85(1.87)			76.98	76.54(2.35)			

Table 6.3: Friedman ranking for BO and hybrid methods with RST measure

Methods	FR	Rank
BO	1.31	1
Hybrid-P	2.25	2
Hybrid-S	2.43	3

6.3.2 BO method with L-FRFS measure

To validate the performance of BO, the results of feature selection obtained using BO algorithm have been compared with results of feature selection using Hybrid-P and Hybrid-S discussed in previous chapter. In this section, results for feature selection using the BO method have been presented using fuzzy rough dependency measure of L-FRFS. Superiority of the BO method, w.r.t Hybrid-P and Hybrid-S methods of feature selection is established through the results shown in Table 6.6 and following discussions.

Tables 6.4 and 6.5 present the results of feature selection using the BO method. The results of the feature selection are presented in Table 6.4 in terms of number of features in the reducts suggested by each of the methods. For the 12 runs of each method for any dataset, best reduct size, mean reduct size and standard deviation (s.d.) among the reduct sizes have been presented.

In Tables 6.4 and 6.5, t-test (a parametric test) and Wilcoxon test (non-parametric test) have also represented. the significance value of 0.05 has been taken in these tests.

It is observed from Tables 6.4 and 6.5 that in the case of *Cleveland*, *Ecoli*, *Glass*, *Ionosphere*, *Soybean small* and *Wine* datasets, BO is as good as other methods, in terms of reduct size and classification accuracy. In the case of *Lung* the t-test as well as Wilcoxon test for reduct size (Table 6.4 and 6.5) suggest that BO performs better than Hybrid-P but have comparable performance w.r.t Hybrid-S methods. In the case of $LSVT$ dataset, the BO method improves the result significantly, in terms of number of features (Table 6.4). For *Soybean-small*, BO gives guaranteed optimal result i.e. reduct size as 2. In almost all the datasets BO has better performance, especially in the cases of *Lung*, *Wine* and$LSVT$ datasets.

Tables 6.4 and 6.5 show best results reported in literature in terms of Mean Subset size (denoted as MSS) and classification accuracy (denoted as CA). Further, It is evident

Table 6.4: BO with L-FRFS measure: Comparison of reduct size with statistical t-test, and Wilcoxon test. [Statistical t-test is denoted as 'S'. Wilcoxon test is denoted as 'WT'. Mean Subset Size is denoted as 'MSS'.]

Dataset (Total features)	Feature Selection Method	Features Subset Size				Best MSS reported in literature
		Best	Mean(s.d.)	S	WT	
Cleveland(13)	Hybrid-P	6	6(0)	-	-	7.81 [30]
	Hybrid-S	6	6(0)	-	-	
	BO	**6**	6(0)			
Ecoli(7)	Hybrid-P	5	5(0)	-	-	3 [52]
	Hybrid-S	5	5(0)	-	-	
	BO	**5**	5(0)			
Glass(9)	Hybrid-P	8	8(0)	-	-	8.44 [30]
	Hybrid-S	8	8(0)	-	-	
	BO	**8**	8(0)			
Ionosphere(34)	Hybrid-P	6	6.16(0.38)	-	-	7.3 [30]
	Hybrid-S	6	6.08(0.28)	-	-	
	BO	**6**	**6.25(0.45)**			
Lung(56)	Hybrid-P	3	4.33(0.65)	-	-	NA
	Hybrid-S	3	4.5(0.79)	-	-	
	BO	**3**	**4.12(0.16)**			
Soybean small(35)	Hybrid-P	2	2(0)	-	-	2 [29]
	Hybrid-S	2	2(0)	-	-	
	BO	**2**	2(0)			
Wine(13)	Hybrid-P	4	4(0)	-	-	2 [52]
	Hybrid-S	4	4(0)	-	-	
	BO	**4**	4(0)			
LSVT(310)	Hybrid-P	8	11.16(1.94)	-	-	NA
	Hybrid-S	9	14.75(2.09)	-	-	
	BO	**7**	**8.5(1.24)**			

from the Tables 1.1, 6.4 and 6.5 that the performance of BO is better than that of state-of-the-art methods suggested in literature for the dataset *Cleveland*, *Glass* and *Ionosphere*. While using fuzzy rough dependency measure as fitness function, these method provides better accuracy for *Ecoli* and *Wine* dataset at the cost of relatively large reduct size than the existing best method reported in literature. BO produces the stable result with zero or very low standard deviation.

Table 6.6 shows the results of Friedman test giving the ranking of the performance of all the methods. It is observed that BO ranks the best as compared to Hybrid-P and Hybrid-S methods.

Table 6.5: BO with L-FRFS measure: Comparison of classification accuracy with statistical t-test, and Wilcoxon test. [Classification accuracy is denoted as 'CA'. Statistical t-test is denoted as 'S'. Wilcoxon test is denoted as 'WT'.]

Dataset (total features)	Feature Selection Method	Classification Accuracy (CA)												Best CA reported in literature
		Classifier: J48				Classifier: JRip				Classifier: PART				
		Best	Mean(s.d.)	S	WT	Best	Mean(s.d.)	S	WT	Best	Mean(s.d.)	S	WT	
Cleveland(13)	Hybrid-P	52.47	52.47(0)	-	-	53.79	53.79(0)	-	-	52.14	52.14(0)	-	-	52.6 [30]
	Hybrid-S	52.47	52.47(0)	-	-	53.79	53.79(0)	-	-	52.14	52.14(0)	-	-	
	BO	52.47	52.47(0)			**53.79**	53.79(0)			52.14	52.14(0)			
Ecoli(7)	Hybrid-P	82.44	82.44(0)	-	-	81.25	81.25(0)	-	-	80.65	80.65(0)	-	-	77.38 [52]
	Hybrid-S	82.44	82.44(0)	-	-	81.25	81.25(0)	-	-	80.65	80.65(0)	-	-	
	BO	**82.44**	82.44(0)			81.25	81.25(0)			80.65	80.65(0)			
Glass(9)	Hybrid-P	64.49	64.49(0)	-	-	69.16	69.16(0)	-	-	68.69	68.69(0)	-	-	65.14 [30]
	Hybrid-S	64.49	64.49(0)	-	-	69.16	69.16(0)	-	-	68.69	68.69(0)	-	-	
	BO	64.49	64.49(0)			**69.16**	69.16(0)			68.69	68.69(0)			
Ionosphere(34)	Hybrid-P	**93.45**	90.45(1.92)	-	-	91.45	89.29(1.52)	-	-	93.45	90.17(1.89)	-	-	86.17 [30]
	Hybrid-S	90.88	89.52(0.84)	-	-	91.74	89.38(1.27)	-	-	91.74	89.14(1.07)	-	-	
	BO	91.45	88.55(1.12)			92.31	90.1(1.1)			91.17	89.8(0.99)			
Lung(56)	Hybrid-P	84.37	72.39(8.08)	v	v	87.5	73.69(8.47)	v	v	81.25	71.61(6.17)	v	v	NA
	Hybrid-S	87.5	79.42(8.88)	-	-	87.5	79.16(8.25)	-	-	87.5	74.73(9.65)	-	-	
	BO	**87.5**	83.67(1.61)			87.5	83.67(1.51)			87.5	81.85(2.6)			
Soybean small(35)	Hybrid-P	100	100(0)	-	-	100	100(0)	-	-	100	100(0)	-	-	100 [29]
	Hybrid-S	100	100(0)	-	-	100	100(0)	-	-	100	100(0)	-	-	
	BO	**100**	100(0)			100	100(0)			100	100(0)			
Wine(13)	Hybrid-P	93.82	93.67(0.48)	-	-	92.13	90.07(0.64)	-	-	93.82	93.67(0.48)	-	-	90.44 [52]
	Hybrid-S	93.82	93.82(0)	-	-	89.89	89.89(0)	-	-	93.82	93.82(0)	-	-	
	BO	**93.82**	93.82(0)			89.89	89.89(0)			93.82	93.82(0)			
LSVT(310)	Hybrid-P	81.75	75.6(3.76)	-	-	80.16	76.42(3.7)	-	-	80.16	75.33(3.79)	-	-	NA
	Hybrid-S	83.33	75.26(6.67)	-	-	81.75	76.52(4.23)	-	-	**84.13**	75.59(6.45)	-	-	
	BO	80.16	74.28(2.27)			80.16	75.25(2.13)			81.74	76.18(3.01)			

Table 6.6: Friedman ranking with L-FRFS measure

Methods	FR	Rank
BO	1.8906	1
Hybrid-S	2.0156	2
Hybrid-P	2.0938	3

6.4 Conclusion

A new feature selection method based on BO has been proposed applying RST and L-FRFS measures. Effectiveness of the BO was established in terms of number of reduct size, classification accuracy, t-test, Wilcoxon test and Friedman test. The BO has shown its effectiveness on large and practical datasets. To demonstrate that the proposed feature selection method BO is successful in identifying the irrelevant and redundant features, classification accuracies have also been evaluated and discussed. It has been shown that the values of the accuracy remain acceptable with the resulting feature selection.

To validate that the accuracy of the reducts obtained applying BO, are acceptable, classification accuracies using three classifiers were obtained. Data mining workbench WEKA [59, 60], has been used to compute classification accuracies for classifiers J48, JRip, and PART. Further, results of parametric test (t-test) and non-parametric test (Wilcoxon test) have also been presented in this chapter and it is observed that in general the BO has comparable accuracy.

Tables 6.3 and 6.6 show the results of the Friedman test, giving the ranking of the performance of all three methods. It is observed that BO ranks the best as compared to Hybrid-P and Hybrid-S methods, in case of RST as well as L-FRFS measures.

As suggested by the Friedman ranking test, Hybrid-P method have better ranking than Hybrid-S with RST measure, but the case is opposite in the case of L-FRFS measure, but in both the cases they have comparable performance, therefore, both should be attempted.