Research Journal of Applied Sciences, Engineering and Technology 10(12): 1389-1395, 2015
DOI: 10.19026/rjaset.10.1839
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: March 25, 2015 Accepted: April 22, 2015 Published: August 25, 2015

Research Article

Multi-Swarm Bat Algorithm

[1]Ahmed Majid Taha, [2]Soong-Der Chen and [3]Aida Mustapha
[1]Soft Computing and Data Mining Center, Universiti Tun Hussein Onn Malaysia
86400 Parit Raja, Batu Pahat, Johor,
[2]College of Information Technology, Universiti Tenaga Nasional, 43000 Kajang, Selangor,
[3]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia,
86400 Parit Raja, Batu Pahat, Malaysia

Abstract: In this study a new Bat Algorithm (BA) based on multi-swarm technique called the Multi-Swarm Bat Algorithm (MSBA) is proposed to address the problem of premature convergence phenomenon. The problem happens when search process converges to non-optimal solution due to the loss of diversity during the evolution process. MSBA was designed with improved ability in exploring new solutions, which was essential in reducing premature convergence. The exploration ability was improved by having a number of sub-swarms watching over the best local optima. In MSBA, when the quality of best local optima does not improve after a pre-defined number of iterations, the population is split equally into several smaller sub-swarms, with one of them remains close to the current best local optima for further exploitation while the other sub-swarms continue to explore for new local optima. The proposed algorithm has been applied in feature selection problem and the results were compared against eight algorithms, which are Ant Colony Optimization (ACO), Genetic Algorithm (GA), Tabu Search (TS), Scatter Search (SS), Great Deluge Algorithm (GDA) and stander BA. The results showed that the MSBA is much more effective that it is able to find new best solutions at times when the rest of other algorithms are not able to.

Keywords: Bat algorithm, bio-inspired algorithms, data mining, feature selection, multi-swarm, optimization

INTRODUCTION

In bio-inspired algorithms, there must exists a balance between exploitation (intensification) and exploration (diversification); two of which very important components that affect the overall efficiency and performance of the particular learning algorithm. In intensification the promising regions are explored more thoroughly in the hope to find better solutions. Diversification non-explored regions must be visited to be sure that all regions of the search space are evenly explored and that the search is not confined to only a reduced number of regions. Too much intensification and too little diversification could cause the system to be trapped in local optima, which could lead to loss of the global optimum. Only a good combination of these two major components will usually ensure that the global optimality is reachable. This is essentially the problem of premature convergence suffered by many bio-inspired algorithms (Dorigo and Blum, 2005; Mallipeddi *et al*., 2011; Choubey and Kharat, 2013), whereby learning stops during convergence of non-optimal solutions due to the loss of diversity in the process of evolution.

Literature has shown that BA, as a bio-inspired algorithm, also it may suffers from the premature convergence phenomenon (Lin *et al*., 2012; Xie *et al*., 2013; Wang *et al*., 2013a, b). In order to avoid this, Wang *et al*. (2013b) enhanced the bats' flight skills as well as flight modes by dynamically regulating the flight speed and direction while assuming the position of the prey continuously change with time. Meanwhile, Lin *et al*. (2012) proposed a synergistic approach based on chaotic sequence and chaotic Levy flight in the meta-heuristic search in efficiently generating new solutions. The study suggested chaotic sequence is able to generate several neighborhoods of suboptimal solutions and maintain the variability in the solutions. This, in turn, could prevent premature convergence.

LITERATURE REVIEW

Premature convergence has also been treated using Multi-Swarm Approach (MSA) (Brits *et al*., 2007; Zhao *et al*., 2008). Multi-swarm approach is based on the use of multiple sub-swarms instead of one standard swarm. The general approach in multi-swarm optimization is that each sub-swarm focuses on a specific region while a specific diversification method

Corresponding Author: Ahmed Majid Taha, Soft Computing and Data Mining Center, Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia

decides where and when to start the sub-swarms. The idea of MSA start with the implementation of parallel Genetic Algorithm called island model, whereby each population is located into a processor (island) with their own independent evolution process. In order to promote cooperation between islands, a new operator, called migration is created. According to some predefined strategy, individuals migrate from one island to another. The island model has been applied in various applications such as multiprocessor scheduling problem (Corcoran and Wainwright, 1994), surveillance tests policy optimization (Pereira and Lapa, 2003), airline crew scheduling (Levine, 1996), multi-track music segmentation (Rafael *et al*., 2013) and many others. Overall, implementation of the island model in previous studies have reported better search performance as compared to single population model.

Like-wise, multi-swarm approaches have shown similar outstanding results as in Muhlenbein (1991), Niu *et al*. (2005), Brits *et al*. (2007) and Marinakis and Marinaki (2010). The research began with application in PSO called the multi-swarm, which was originally presented by Parsopoulos and Vrahatis (2002) and later developed by Blackwell and Branke (2004). The main idea is to extend the single population in PSO into charged particle swarm optimization methods by constructing interacting multi-swarms. Niu *et al*. (2005) introduced a master-slave model as inspired by the phenomenon of symbiosis in natural ecosystem. In a master-slave model, the population consists of one master swarm and several slave swarms. The slave swarms execute PSO or its variants independently to maintain the diversity of particles, while the master swarm enhances its particles based on its own knowledge and also the knowledge of the particles in the slave swarms.

Several variations of multi-swarm PSO have been employed across different domains such as dynamic optimization (Blackwell and Branke, 2004; Liang and Suganthan, 2006; Liu *et al*., 2012) and benchmarked function (Niu *et al*., 2005; Liang and Suganthan, 2006). The reported results for these studies demonstrate superior performance of multi-swarm PSO as compared to the single-swarm PSO. Nonetheless, in the literature, multi-swarm algorithm has only been applied to the problem of feature selection by Liu *et al*. (2011), which is the Multi-Swarm PSO. To date, multi-Swarm strategy has not been considered for Bat Algorithm yet, hence motivating this research to develop a multi-swarm BA that is capable avoiding premature convergence phenomenon in addition to improving exploration and exploitation ability of BA.

METHODOLOGY

Proposed multi-swarm bat algorithm: Swarm algorithms are founded based on the theory that similar animals aggregate together, generally cruising in the same direction. For instance, termites swarm to build colonies, birds swarm to migrate and bees swarm to collect food. Multi-swarm algorithms are based on the use of multiple sub-swarms instead of one swarm. In general, each sub-swarm will focus on specific region, therefore each swarm will learn from its experience in searching for the global optima. Swarm algorithms are suitable for problems where multiple local optima exist. In the BA, the searching space for promising areas is identified by the loudness and the pulse rate in bats. So in some cases the bats may mislead, because they start to slow down the speed assuming they are reaching global optimal solution area while in reality they are not. Therefore the bats may get stuck in local optima, this behavior of bats will result early convergence to non-optimal solution.

For applying single swarm or stander BA in feature selection problem it can be referred to Taha and Tang (2013) and Taha *et al*. (2013). In this study, implementation of the MSBA starts with single swarm that consists of a group of M bats looking for global optima. When the best solution cannot be improved after certain number of iterations, all the bats will be re-initialized randomly except the one with the best solution which will remain for further exploitation. The bats will then be grouped into S new and independent sub-swarms.

All swarms are autonomous as they have their own independent evolution process, such as the global best and the average loudness. The new swarms will discover different region of the search space looking for better solutions. If any of the sub-swarm finds a new solution better than the previous global best solutions from other swarms, the best swarm will call bats from other swarms. In this case all bats will migrate to the swarm with global best solution, each with their own loudness, pulse rates and velocity. Traveling from the old positions to the new potential positions facilitate exploration of new areas in the search space because the bats communicate with the new global best. Furthermore, crossing in the features between the new global best and the old global best provide unique solutions that single population cannot provide. If any bat from the immigrants' bats finds a new solution better than the previous potential swarm solution, the new solution can be considered as the new potential swarm and the process continues as illustrated in Fig. 1 with the union and successive movements.

After the new global optima have been recognized, all bats will facilitate more exploitation in the proposed new swarm. In the case where even the multi-swarm search fails to improve the global best solution after T iterations, then all the bats will be re-initialized except the one with global best solution. The search will then continue in form of single-swarm. The switching between single-swarm and multi-swarm search will continue until the stopping criteria are met.

The destroying and reinitialize sub-swarms serves to enhance the diversification or exploration in the search space and avoid premature convergence, while calls sub-swarms to grouping in single-swarm search

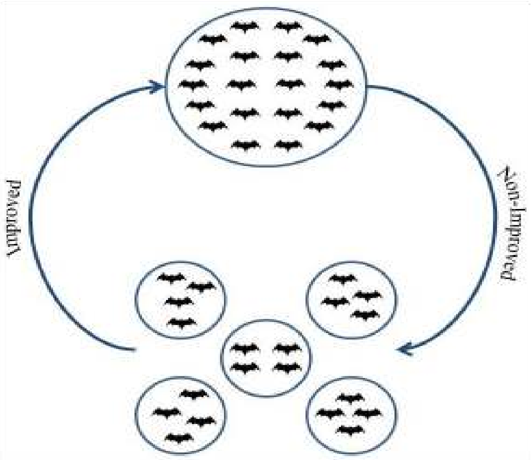

Fig. 1: Multi-swarm movement

aims to enhance the local search or exploitation. The stopping criteria that have been set in this study are as follows; whether the algorithm reached the maximum number of iteration or single feature with maximum number of fitness function, the pseudo code is provided in below.

Pseudo code of multi-swarm bat algorithm:

```
Initialize parameters
Initialized M bats
While stop condition not met
    While the fitness of global best solution is improving before N iteration and stop condition is not met
        Perform single swarm search with M bats
    End-while
    If stop condition is not met
        Keep only the bat with global best solution and initialize the remaining M-1 bats
        Divide the M bats into S swarm
        While the fitness of global best solution is not
            improving before T iteration and stop condition is not met
            Perform single swarm search for each of the S swarms
        End-While
        If new better solution found
            Set global best solution to the new better solution
    Else
        Keep only the bat with global best solution and initialize the remaining M-1 bats
    End-if
  End-if
End-while
```

EXPERIMENTAL RESULTS

In order to demonstrate the effectiveness algorithm, a series of experiments were carried out on twelve benchmarked datasets from various domains have been selected. Each dataset has different number of features and samples as shown in Table 1. The parameters is set as follow *S* equal to 4, *M* equal to 50, *N* equal to 30, *T* equal to 60. The experimental results were compared against seven well-known optimization algorithms as listed below:

Table 1: Characteristics of datasets

Datasets	No. of features	No. of samples
Lung	56	32
WQ	38	521
Derm2	34	358
Derm	34	366
LED	24	2000
Mushroom	22	8124
Credit	20	1000
Vote	16	300
Heart	13	294
Exactly2	13	1000
Exactly	13	1000
M-of-N	13	1000

- Genetic algorithm for Rough Set Attribute Reduction (GenRSAR) (Jensen and Shen, 2003)
- Ant colony optimization for Rough Set Attribute Reduction (AntRSAR) (Jensen and Shen, 2003)
- Simulated annealing for Rough Set Attribute Reduction (SimRSAR) (Jensen and Shen, 2004)
- Tabu Search for Attribute Reduction (TSAR) (Hedar *et al*., 2008)
- Scatter Search for Attribute Reduction (SSAR) (Jue *et al*., 2009)
- Scatter Search for Attribute Reduction (SSAR) (Jue *et al*., 2009)
- Great Deluge algorithm for Rough Set Attribute Reduction (GD-RSAR) (Abdullah and Jaddi, 2010)
- Bat Algorithm for Attribute Reduction (BAAR) (Taha and Tang, 2013)

In this study MSBA is based on rough set theory similar to the above mentioned algorithms, the experimental setups were also the same as provided in the literature. For each dataset, MSBA was run 20 times with different initial solutions and terminated after 250 iterations. All algorithms have the same number of runs for each dataset, except the results of SimRSAR using Heart, Vote and Drem2 datasets, for which the number of runs are 30, 30 and 10, respectively. Comparison on the number of features is summarized in Table 2 and 3, note that between the brackets is indicates the total number of runs that this cardinality was achieved. The number of features without brackets denotes that the method could obtain this number of features for all runs.

Next the results were then statistically tested using two tests; Kolmogorov-Smirnov and Levene test (Lilliefors, 1967). However, the Kolmogorov-Smirnov and Levene test did not meet the assumptions of

Table 2: Number of features comparison (a)

Datasets	MSBA	BAAR	SimRSAR	AntRSAR	GenRSAR
M-of-N	$4^{(2)}$ $5^{(7)}$ $6^{(11)}$	6	6	6	$6^{(6)}$ $7^{(12)}$
Exactly	$4^{(1)}$ $5^{(2)}$ $6^{(17)}$	6	6	6	$6^{(10)}$ $7^{(10)}$
Exactly2	$7^{(2)}$ $8^{(2)}$ $9^{(5)}$ $10^{(11)}$	10	10	10	$10^{(9)}$ $11^{(11)}$
Heart	$4^{(3)}$ $5^{(17)}$	5	$6^{(29)}$ $7^{(1)}$	$6^{(18)}$ $7^{(2)}$	$6^{(18)}$ $7^{(2)}$
Vote	$5^{(1)}$ $6^{(1)}$ $7^{(6)}$ $8^{(12)}$	8	$8^{(15)}$ $9^{(15)}$	8	$8^{(2)}$ $9^{(18)}$
Credit	$7^{(8)}$ $8^{(12)}$	8	$8^{(18)}$ $9^{(1)}$ $11^{(1)}$	$8^{(12)}$ $9^{(4)}$ $10^{(4)}$	$10^{(6)}$ $11^{(14)}$
Mushroom	$3^{(1)}$ $4^{(19)}$	4	4	4	$5^{(1)}$ $6^{(5)}$ $7^{(14)}$
LED	$4^{(2)}$ $5^{(18)}$	5	5	$5^{(12)}$ $6^{(4)}$ $7^{(3)}$	$6^{(1)}$ $7^{(3)}$ $8^{(16)}$
Derm	$5^{(1)}$ $6^{(17)}$ $7^{(2)}$	$6^{(13)}$ $7^{(7)}$	$6^{(12)}$ $7^{(8)}$	$6^{(17)}$ $7^{(3)}$	$10^{(6)}$ $11^{(14)}$
Derm2	$7^{(1)}$ $8^{(12)}$ $9^{(7)}$	$9^{(12)}$ $10^{(8)}$	$8^{(3)}$ $9^{(7)}$	$8^{(3)}$ $9^{(17)}$	$10^{(4)}$ $11^{(16)}$
WQ	$11^{(1)}$ $12^{(5)}$ $13^{(14)}$	$12^{(2)}$ $13^{(11)}$ $14^{(7)}$	$13^{(16)}$ $14^{(4)}$	$12^{(2)}$ $13^{(7)}$ $14^{(11)}$	16
Lung	$4^{(11)}$ $5^{(9)}$	$4^{(10)}$ $5^{(6)}$ $6^{(4)}$	$4^{(7)}$ $5^{(12)}$ $6^{(1)}$	4	$6^{(8)}$ $7^{(12)}$

Table 3: Number of features comparison (b)

Datasets	MSBA	TSAR	GD-RSAR	SSAR
M-of-N	$4^{(2)}$ $5^{(7)}$ $6^{(11)}$	6	$6^{(10)}$ $7^{(10)}$	6
Exactly	$4^{(1)}$ $5^{(2)}$ $6^{(17)}$	6	$6^{(7)}$ $7^{(10)}$ $8^{(3)}$	6
Exactly2	$7^{(2)}$ $8^{(2)}$ $9^{(5)}$ $10^{(11)}$	10	$10^{(14)}$ $11^{(6)}$	10
Heart	$4^{(3)}$ $5^{(17)}$	6	$9^{(4)}$ $10^{(16)}$	6
Vote	$5^{(1)}$ $6^{(1)}$ $7^{(6)}$ $8^{(12)}$	8	$9^{(17)}$ $10^{(3)}$	8
Credit	$7^{(8)}$ $8^{(12)}$	$8^{(13)}$ $9^{(5)}$ $10^{(2)}$	$11^{(11)}$ $12^{(9)}$	$8^{(9)}$ $9^{(8)}$ $10^{(3)}$
Mushroom	$3^{(1)}$ $4^{(19)}$	$4^{(17)}$ $5^{(3)}$	$4^{(8)}$ $5^{(9)}$ $6^{(3)}$	$4^{(12)}$ $5^{(8)}$
LED	$5^{(18)}$ $4^{(2)}$	5	$8^{(14)}$ $9^{(6)}$	5
Derm	$5^{(1)}$ $6^{(17)}$ $7^{(2)}$	$6^{(14)}$ $7^{(6)}$	$12^{(14)}$ $13^{(6)}$	6
Derm2	$7^{(1)}$ $8^{(12)}$ $9^{(7)}$	$8^{(2)}$ $9^{(14)}$ $10^{(4)}$	$11^{(14)}$ $12^{(6)}$	$8^{(2)}$ $9^{(18)}$
WQ	$11^{(1)}$ $12^{(5)}$ $13^{(14)}$	$12^{(1)}$ $13^{(13)}$ $14^{(6)}$	$15^{(14)}$ $16^{(6)}$	$13^{(4)}$ $14^{(16)}$
Lung	$4^{(11)}$ $5^{(9)}$	$4^{(6)}$ $5^{(13)}$ $6^{(1)}$	$4^{(5)}$ $5^{(2)}$ $6^{(13)}$	4

Table 4: Wilcoxon test results

	BAAR-MSBA	SimRSAR-MSBA	AntRSAR-MSBA	GenRSAR-MSBA
M-of-N	0.005 (MSBA)	0.005 (MSBA)	0.005 (MSBA)	0.000 (MSBA)
Exactly	0.102	0.102	0.102	0.000 (MSBA)
Exactly2	0.007 (MSBA)	0.007 (MSBA)	0.007 (MSBA)	0.000 (MSBA)
Heart	0.083	0.000 (MSBA)	0.000 (MSBA)	0.000 (MSBA)
Vote	0.008 (MSBA)	0.001 (MSBA)	0.008 (MSBA)	0.000 (MSBA)
Credit	0.005 (MSBA)	0.002 (MSBA)	0.000 (MSBA)	0.000 (MSBA)
Mushroom	0.317	0.317	0.317	0.000 (MSBA)
LED	0.157	0.157	0.010 (MSBA)	0.000 (MSBA)
Derm	0.014 (MSBA)	0.008 (MSBA)	0.157	0.000 (MSBA)
Derm2	0.000 (MSBA)	-	0.001 (MSBA)	0.000 (MSBA)
WQ	0.001 (MSBA)	0.002 (MSBA)	0.000 (MSBA)	0.000 (MSBA)
Lung	0.025 (MSBA)	0.025 (ANT)	0.003 (MSBA)	0.000 (MSBA)

Table 5: Wilcoxon test results

	TSAR-MSBA	GD-RSAR-MSBA	SSAR-MSBA
M-of-N	0.005 (MSBA)	0.000 (MSBA)	0.005 (MSBA)
Exactly	0.102	0.000 (MSBA)	0.102
Exactly2	0.007 (MSBA)	0.000 (MSBA)	0.007 (MSBA)
Heart	0.000 (MSBA)	0.000 (MSBA)	0.000 (MSBA)
Vote	0.008 (MSBA)	0.000 (MSBA)	0.008 (MSBA)
Credit	0.000 (MSBA)	0.000 (MSBA)	0.000 (MSBA)
Mushroom	0.102	0.001 (MSBA)	0.007 (MSBA)
LED	0.157	0.000 (MSBA)	0.157
Derm	0.025 (MSBA)	0.000 (MSBA)	0.564
Derm2	0.000 (MSBA)	0.000 (MSBA)	0.001 (MSBA)
WQ	0.001 (MSBA)	0.000 (MSBA)	0.000 (MSBA)
Lung	0.014 (MSBA)	0.000 (MSBA)	0.003 (ANT)

normality distribution and equality of variance, which then led us to the use of Wilcoxon test. Table 4 and 5 presents Wilcoxon test results for the proposed MSBA algorithm against other feature selection algorithms, between the brackets refer to the algorithm that performs better than another algorithm. The classification accuracy for the resulted features achieved by MSBA have been evaluated using three deferent classifier which are JRip, PART and J48. Then the results compared with BAAR, SimRSAR and GenRSAR, the average classification accuracies in percentage are presented in Table 6 to 8.

Table 6: Average classification accuracy using JRip classifier

Datasets	MSBA	BAAR	SimRSAR	GenRSAR
M-of-N	94.24	98.90	98.90	98.90
Exactly	93.02	99.30	99.30	99.31
Exactly2	74.22	73.80	73.80	73.80
Heart	66.96	66.32	78.74	78.57
Vote	94.93	95.00	95.00	94.66
Credit	69.35	69.78	69.95	70.57
Mushroom	99.97	99.97	99.98	99.98
LED	100.00	100.00	100.00	100.00
Derm	65.70	70.78	68.46	80.54
Derm2	79.63	82.09	82.28	80.49
WQ	66.18	65.73	66.02	64.81
Lung	70.62	75.30	71.55	68.74

Table 7: Average classification accuracy using PART classifier

Datasets	MSBA	BAAR	SimRSAR	GenRSAR
M-of-N	95.20	100.00	100.00	100.00
Exactly	94.02	100.00	100.00	99.98
Exactly2	76.20	78.30	78.30	78.30
Heart	65.60	65.30	75.81	75.06
Vote	94.13	94.00	94.00	94.19
Credit	70.21	70.60	71.35	71.88
Mushroom	100.00	100.00	100.00	100.00
LED	100.00	100.00	100.00	100.00
Derm	61.60	63.60	61.90	79.94
Derm2	82.29	84.20	84.68	83.85
WQ	62.97	64.27	62.33	62.24
Lung	71.25	70.62	75.93	65.93

Table 8: Average classification accuracy using J48 classifier

Datasets	MSBA	BAAR	SimRSAR	GenRSAR
M-of-N	95.12	100.00	100.00	100.00
Exactly	93.72	100.00	100.00	99.92
Exactly2	74.18	73.10	73.10	73.10
Heart	62.98	62.92	77.68	77.21
Vote	94.06	94.00	94.00	93.93
Credit	70.41	70.68	70.83	71.08
Mushroom	100.00	100.00	100.00	100.00
LED	100.00	100.00	100.00	100.00
Derm	64.17	70.97	66.03	80.37
Derm2	83.10	84.96	84.74	84.29
WQ	64.04	65.02	62.83	64.35
Lung	70.31	72.49	75.93	67.80

DISCUSSION

The experimental results showed that MSBA has a superior performance in eleven out of twelve datasets. This implies that the proposed algorithm is able to find the new best known solutions that the other algorithms fail to search. In the Exactly2 dataset, MSBA found new three best known solutions with slightly higher classification accuracy in two classifiers, which are JRip and J48. MSBA is also significantly better than the single swarm BA as well as other algorithms as shown in Table 4 and 5. In the Mushroom dataset, even though the classification accuracy achieved by MSBA is the same with other algorithms, but MSBA has the advantage of new found solutions and it is statistically better than GenRSAR and GD-RSAR.

In the Led dataset, MSBA achieved 100% classification accuracy aside from obtaining a new best known solution. In the Credit dataset, it can be noted that MSBA obtained the new best known solution with seven features in eight runs out of twenty runs. It can also be noted that MSBA is significantly better than all other algorithms. For the Credit dataset, all algorithms provided almost the same classification accuracy in all three classifiers. In the Vote dataset, MSBA found three new best known solutions with the same classification accuracy as other classifiers. In particular, the results for MSBA are significantly better than BAAR and other traditional methods.

The Derm2 and WQ dataset are considered as the most challenging datasets as no algorithm is able to obtain best solution for all runs as can be seen in Table 2 and 3. Nonetheless, MSBA still performed significantly better than all others algorithm with the same accuracy in the WQ dataset. Meanwhile, in the Exactly and Derm datasets, although the classification accuracy achieved by MSBA decreased moderately as compared to other method, but MSBA was still able to find new best solutions. For the Heart dataset, the results provided by MSBA with regards to the classification accuracy were the same with those delivered by BAAR but lower than other methods. Finally in the Lung dataset, MSBA could not find any new best known and the classification accuracy solution was also lower than average of the other algorithm results. Nevertheless, the proposed MSBA is still significantly better than BAAR and the other four others method (GenRSAR, AntRSAR, TSAR and GD-RSAR).

As the conclusion, the proposed MSBA is significantly better than other single-swarm algorithms in eight out of twelve datasets. It also outperformed other algorithms in most of the datasets. MSBA is able obtain the lowest number of features across the most data sets and the results also showed that MSBA is able to find new best solutions while maintaining a competitive classification accuracy. Nevertheless, in few cases, the accuracy results obtained by MSBA are less satisfactory than those excellent ones reported by rest three methods which are BAAR, SimRSAR and GenRSAR. This is attributed to the use of rough set theory, whereby MSBA has less sensitivity to the changes in the prediction accuracy. The superior performance of MSBA and the ability to find new solution is influenced by the multi-swarm movement in discovering new areas within the search space, beyond the reach of other classical algorithms.

CONCLUSION

MSBA is proposed to address the problem of premature convergence phenomenon that usually occurs in BA. The problem happens when search process converges to non-optimal solution due to the loss of diversity during the evolution process. MSBA was designed with improved ability in exploring new solutions, which was essential in reducing premature convergence. In MSBA, when the quality of best local optima does not improve after a pre-defined number of iterations, the population is split equally into several smaller sub-swarms, with one of them remains close to the current best local optima for further exploitation while the other sub-swarms continue to explore for new local optima. A series of experiments were carried out and the results of MSBA were compared against the eight conventional methods. MSBA achieved very good results, whereby it produced new best solutions in 11 out of 12 cases. Furthermore, the statistical analysis showed that the proposed method in most data sets performed significantly better than the other methods in terms of selecting smaller subset of features with providing competitive prediction accuracy. For future work is highly recommended to extensive study of the applications of proposed Multi-Swarm BA model in other complex practical optimization problems is necessary, to fully investigate the properties, evaluate and optimize MSBA performance.

REFERENCES

Abdullah, S. and N. Jaddi, 2010. Great Deluge Algorithm for Rough Set Attribute Reduction. In: Zhang, Y., A. Cuzzocrea, J. Ma, K.I. Chung, T. Arslan and X. Song, (Eds.), Database Theory and Application, Bio-science and Bio-technology. Springer, Berlin, Heidelberg, ISBN: 3642176224, pp: 310.

Blackwell, T. and J. Branke, 2004. Multi-swarm Optimization in Dynamic Environments. In: Raidl, G., S. Cagnoni, J. Branke, D. Corne, R. Drechsler *et al.* (Eds.), Applications of Evolutionary Computing. Springer, Berlin, Heidelberg.

Brits, R., A.P. Engelbrecht and F. Van Den Bergh, 2007. Locating multiple optima using particle swarm optimization. Appl. Math. Comput., 189: 1859-1883.

Choubey, N.S. and M.U. Kharat, 2013. Hybrid system for handling premature convergence in GA - Case of grammar induction. Appl. Soft Comput., 13: 2923-2931.

Corcoran, A.L. and R.L. Wainwright, 1994. A parallel island model genetic algorithm for the multiprocessor scheduling problem. Proceeding of the ACM Symposium on Applied Computing. Phoenix, Arizona, USA.

Dorigo, M. and C. Blum, 2005. Ant colony optimization theory: A survey. Theor. Comput. Sci., 344: 243-278.

Hedar, A.R., J. Wang and M. Fukushima, 2008. Tabu search for attribute reduction in rough set theory. Soft Comput. Fusion Found. Methodol. Appl., 12: 909-918.

Jensen, R. and Q. Shen, 2003. Finding rough set reducts with ant colony optimization. Proceeding of the UK Workshop on Computational Intelligence, pp: 15-22.

Jensen, R. and Q. Shen, 2004. Semantics-preserving dimensionality reduction: Rough and fuzzy-rough-based approaches. IEEE T. Knowl. Data En., 16: 1457-1471.

Jue, W., A.R. Hedar, Z. Guihuan and W. Shouyang, 2009. Scatter search for rough set attribute reduction. Proceeding of the International Joint Conference on Computational Sciences and Optimization, pp: 531-535, April 24-26.

Levine, D., 1996. A parallel genetic algorithm for the set partitioning problem. In: Osman, I. and J. Kelly (Eds.), Meta-heuristics: Theory and Applications. Kluwer Academic Publishers, Boston, MA, USA, pp: 23-35.

Liang, J. and P. Suganthan, 2006. Dynamic multi-swarm particle swarm optimizer with a novel constraint-handling mechanism. Proceeding of IEEE Congress on Evolutionary Computation (CEC, 2006), pp: 9-16.

Lilliefors, H.W., 1967. On the kolmogorov-smirnov test for normality with mean and variance unknown. J. Am. Stat. Assoc., 62: 399-402.

Lin, J.H., C.W. Chou, C.H. Yang and H.L. Tsai, 2012. A chaotic levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems. J. Comput. Inform. Technol., 2(2): 56-63.

Liu, J., X. Ren and H. Ma, 2012. Adaptive swarm optimization for locating and tracking multiple targets. Appl. Soft Comput., 12: 3656-3670.

Liu, Y., G. Wang, H. Chen, H. Dong, X. Zhu and S. Wang, 2011. An improved particle swarm optimization for feature selection. J. Bionic Eng., 8: 191-200.

Mallipeddi, R., P.N. Suganthan, Q.K. Pan and M.F. Tasgetiren, 2011. Differential evolution algorithm with ensemble of parameters and mutation strategies. Appl. Soft Comput., 11: 1679-1696.

Marinakis, Y. and M. Marinaki, 2010. A hybrid multi-swarm particle swarm optimization algorithm for the probabilistic traveling salesman problem. Comput. Oper. Res., 37: 432-442.

Muhlenbein, H., 1991. Evolution in Time and Space-the Parallel Genetic Algorithm. In: Foundations of Genetic Algorithms. Morgan Kaufmann Publishers Inc., San Farncisco, USA, pp: 316-337.

Niu, B., Y. Zhu and X. He, 2005. Multi-population Cooperative Particle Swarm Optimization. In: Capcarrère, M., A. Freitas, P. Bentley, C. Johnson and J. Timmis, (Eds.), Advances in Artificial Life. Springer, Berlin, Heidelberg.

Parsopoulos, K.E. and M.N. Vrahatis, 2002. Particle swarm optimization method in multiobjective problems. Proceeding of the ACM Symposium on Applied Computing. ACM, Madrid, Spain.

Pereira, C.M.N.A. and C.M.F. Lapa, 2003. Parallel island genetic algorithm applied to a nuclear power plant auxiliary feedwater system surveillance tests policy optimization. Ann. Nucl. Energy, 30: 1665-1675.

Rafael, B., M. Affenzeller and S. Wagner, 2013. Application of an Island Model Genetic Algorithm for a Multi-track Music Segmentation Problem. In: Machado, P., J. Mcdermott and A. Carballal, (Eds.), Evolutionary and Biologically Inspired Music, Sound, Art and Design. Springer, Berlin, Heidelberg.

Taha, A.M. and A.Y.C. Tang, 2013. Bat algorithm for rough set attribute reduction. J. Theor. Appl. Inform. Technol., 51(1).

Taha, A.M., A. Mustapha and S.D. Chen, 2013. Naive bayes-guided bat algorithm for feature selection. Sci. World J., 2013: 1-9, Article ID 325973.

Wang, W., Y. Wang and X. Wang, 2013a. Bat Algorithm with Recollection. In: Huang, D.S., K.H. Jo, Y.Q. Zhou and K. Han, (Eds.), Intelligent Computing Theories and Technology. Springer, Berlin, Heidelberg.

Wang, X., W. Wang and Y. Wang, 2013b. An Adaptive Bat Algorithm. In: Huang, D.S., K.H. Jo, Y.Q. Zhou and K. Han, (Eds.), Intelligent Computing Theories and Technology. Springer, Berlin, Heidelberg.

Xie, J., Y. Zhou, and H. Chen, 2013. A novel bat algorithm based on differential operator and lévy flights trajectory. Comput. Intell. Neurosci., 2013(2013): 13, Article ID 453812.

Zhao, S.Z., J.J. Liang, P.N. Suganthan and M.F. Tasgetiren, 2008. Dynamic multi-swarm particle swarm optimizer with local search for Large Scale Global Optimization. Proceeding of the IEEE World Congress on Computational Intelligence, Evolutionary Computation. Hong Kong, pp: 3845-3852, June 1-6.