IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 6, June 2016

Genetic Algorithms for Classification Rule Discovery: Issues and Challenges

Pooja Goyal[1], Saroj [2]

Research Scholar, Department of Computer Science and Engineering, GJU S&T, Hisar, India[1]

Professor, Department of Computer Science and Engineering, GJU S&T, Hisar, India[2]

Abstract: Classification Rule discovery is one of the major tasks of data mining. Genetic algorithms (GAs) for discovery of classification rules have recently gained increased importance because these are capable of finding global optimal solutions in large and complex search spaces like classification rule discovery. Building a rule based classification model, particularly for large data sets, using Genetic Algorithms becomes a very complex and time-consuming task. This paper enlists some of important issues that influence the efficacy and/or efficiency of GAs in the domain of building classification models. The paper also emphasizes the research direction to make application of GAs more amiable for discovering classification rules.

Keywords: Genetic Algorithm, Data Mining, Classification, Rule Mining

I. INTRODUCTION

Classification is a well-studied data mining task to learn a mapping function (model) between attributes values (designated as predicting attributes) and values of a specially designated class or goal or target attribute of a given dataset [1] [2]. The model is learnt from a sample training data and used to predict the class of unseen data instances by the classification algorithm. Some of the important algorithms to discover classification rules are decision trees, Bayesian classifiers, neural networks (NN), support vector machine (SVM) and rough sets etc. [3], [4], [5]. Some of these algorithms like NN and SVM generate classification models with high accuracy but the structure of these models is complex and not very understandable. Rough sets produce a large number of classification rules that make the classifier incomprehensible for its users. Bayesian classifiers are modelled under the unreasonable assumption of attribute independence and therefore ignore attribute interactions. Decision trees generate accurate and comprehensible classifiers (in the form of (if-then) rules but these build the model by selecting one attribute at a time following some greedy heuristic to decide the relevance of the attributes[5] [6]. Decision tree algorithms largely ignore attribute interactions that may lead to a suboptimal classification model.

Genetic Algorithm (GA) based classifiers have been proposed as alternative classification methods. GAs are stochastic algorithms and their search methods model the natural phenomenon of genetic inheritance and Darwin theory of starvation [7], [8]. Recently, Genetic Algorithms have been extensively applied in the domain of comprehensible and interesting classification rule [9], [10], [11], [12], [13], [31], [32]. Genetic Algorithm based approaches are able to learn good classifiers because of their robust search mechanism in the candidate rule search space. In general, the main purpose for using GAs for classification is that they perform a global search and deal with attribute interactions better than the other greedy rule induction algorithms. However, GAs are not without limitations in the domain of discovery of classification RULES.

The limitations include local convergence and unacceptable long running time due to computationally expensive fitness evaluations, which are to be carried out generations over generations. The high computational cost related with fitness evaluations result in unreasonably long running time and prevents the use of GAs for real time applications [3]. This paper brings forth the issues in applying GAs for discovery of classification rules, how these issues have been addressed and the challenges that are yet to be dealt with.

II. GENETIC ALGORITHMS IN CLASSIFICATION RULE DISCOVERY

The Genetic classifier falls into two categories based on how chromosomes or individuals are encoded. These two approaches are known as Michigan approach and Pittsburgh approach [2]. In the Pittsburgh approach, each individual represents a set of rules representing classification model whereas in the Michigan approach each individual encodes a single rule, a part of classification algorithm. In Pittsburgh approach, fitness computation takes account of attribute as well as rule interactions because it evaluates quality of a whole rule set instead of evaluating each rule in isolation[15], [16]. However, when the search space is large, the optimal solution is difficult to find using this approach [4]. In addition, we have to modify the standard genetic operators to cope with longer and complex individual solutions [11]. Some of the earlier Genetic rule miners that use Pittsburgh approach are Genetic based Inductive Learning (GIL) and High Dimensionality Pattern Discovery and Classification System (HDPDCS). In contrast to Pittsburgh approach, the individuals are simple and shorter in case of Michigan

DOI 10.17148/IJARCCE.2016.56110

approach. Hence, it not only takes less time in computing fitness but also the traditional genetic operators can directly be used [11]. Nevertheless, the problem is that it measures the quality of each rule separately, so it is difficult to evaluate the quality of the rule set as a whole. In other words, the problem of rule interaction is not addressed. In fact, it produces a set of best rules but not the best set of rules [15]. Another major problem with the Michigan approach is that during evolution, rules within the population compete and high fitness rules are selected to generate new offspring rules [15]. This makes a standard genetic algorithm to converge to the single best individual (single best rule). Hence, to foster the discovery of a best set of rules, some kind of niching and speciation techniques have to be applied. Some of the initial GAs following the Michigan approach are Coverage based Genetic Induction (COGIN) and Relational Genetic Algorithm Learner (REGAL). Classification Rule Mining using a GA contains a population initialization module (using either Michigan or Pittsburgh approach), fitness computation and appropriate genetic operators to generate accurate, comprehensible and interesting classification rules. Some of the recent applications of GAs for discovering Classifications rules using Michigan approach and Pittsburgh approaches are [3], [4] and [15], [16] respectively. The researchers have observed that GAs following Pittsburgh approach produce classifiers with high predictive accuracy but these take longer running time.

III. ISSUES AND CHALLENGES

The success of a genetic algorithm depends on achieving a right balance between exploration and exploitation of the underlying search space for a given problem. A genetic algorithm needs to explore the search space so that no area of the search space is under-sampled and so that it can exploit the promising regions to arrive at the global optimal solution. Without achieving the balance between exploration and exploitation, a genetic algorithm is bound to converge to local optimal solutions. This balance is influenced by many factors and is not easy to achieve. Moreover, unreasonably long running time prevents us to apply GA in real time settings. This section lists some of the issues and challenges in designing and applying GA for discovering classification rules.

A. Parameter Setting

It is very difficult to set values of GA parameters such as chromosome length, population size, crossover probability, mutation rate and total number of generations. In any case, individual representation and search operators have bias and effectiveness of any bias is strongly problem dependent. Therefore, there is no uniform parameter setting across the domain of problems and the parameter values influence convergence properties of genetic algorithms [24]. Values of these parameters are fixed based either on previous experience or through experimental tuning before the actual and the final run of a GA. The parameter settings for discovery of classification rules using GA depend on the properties of the underlying dataset. Therefore, the GA parameters may need tuning for each dataset separately. Deterministic control and adaptation of the values of parameters to a specific application have also been used to find out values of GA parameters. In deterministic control, values of a genetic parameter vary by some deterministic rule during a GA run. Adaptation of parameters allows changing values of parameters during GA run based on some performance indicators in preceding generation(s). In the adaptive genetic algorithm (AGA), the probabilities of crossover and mutation are changed depending on the fitness of the solutions. High-fitness solutions are protected and solutions having sub average fitness are disrupted. Hence, there is no need to specify crossover and mutation probabilities [35]. In self-adaptation, the operator settings are encoded into each individual of the population that evolves values of parameters during the GA run [14], [25], [36]. In [37] authors propose a Self-Adaptive Migration Model GA (SAMGA), where parameters such as population size, crossover points and mutation probability for each population are adaptively fixed. In addition, the migration of individuals between populations is decided dynamically. Setting GA parameters through experimental tuning is not adequate and it is a research issue of immense importance for bringing in uniformity in the experimentation. Further research is required to design novel schemes for adapting GA parameters to match the evolutionary state of a Genetic Algorithm.

B. Local Convergence

A GA is expected to converge to a single-most fit peak within a fitness landscape but this does not happen always. A GA may converge to local optima, populating the incorrect peak due to sampling errors. To prevent premature convergence, one needs to modify selection pressure caused by fitness proportionate selection methods and allow the population to maintain diversity during the search. A common answer to this problem is to adjust the GA parameters that control trade-off between exploration and exploitation [17].

On the one hand, in the initial GA runs, a few super individuals with high selection pressure dominate and spread rapidly in the subsequent generations. On the other hand, all the individuals in GA population have approximately same selection pressure near convergence that renders the fitness proportionate selection ineffective. Therefore, fitness scaling is necessary to avoid local convergence. However, these methods enhance diversity by adjusting selection pressure but not able to abolish the problem of premature convergence completely. This has directed the research towards the various parallel implementations of Genetic Algorithms that are able to achieve a better balance between exploration and exploitation of the search space [19], [28]. Researchers need to focus on ways to avoid local convergence through multi-pronged and hybrid approaches since no single approach can completely alleviate the problem of local convergence.

DOI 10.17148/IJARCCE.2016.56110

C. Computationally Expensive Fitness Evaluation

GAs have excellent potential to find optimal solutions in reasonable time but when these are applied to complex problems with large search spaces, there is a significant increase in the time needed to reach reasonably acceptable solutions. The increase in time is mainly due to computationally expensive fitness evaluations. For instance, while applying a Genetic Algorithm for discovery of Classification Rules, it has to perform a complete scan of training database for evaluating the fitness of a candidate solution. A GA implementation sometimes requires several thousands of fitness evaluations over successive generations. A database scan, particularly for large databases, is a computationally expensive operation.

One of the approaches for speeding up a GA is to use only a smaller sample of instances as training data for fitness evaluation[17]. Such an approach can speed up the process only if the sampled training data is representative sample of the whole data. Otherwise, it can reduce the quality of solutions that are arrived, i.e., for discovery of classification rules, it can lead to a classifier with lower predictive accuracy. Another promising method is to use Parallel Genetic Algorithms [18], [28], [34].The basic idea behind most parallel programs is to partition a task into chunks and these chunks are processed concurrently using multiple processors. Application of Master Slave GAs can enhance efficiency by doing fitness computations in parallel on many processors. Island or distributed PGAs have also been used to bring efficacy and efficiency for computationally expensive data mining problems. In case of Island model, we divide the population into few sub-populations and the GA processes each of them independently and simultaneously. The migration operator exchanges the genetic material among the sub-populations [19].However, advanced parallel hardware and programming skills are required to implement PGAs, which are not easily available. Some authors have attempted to augment a GA with a long-term memory in the form of data structures like Binary Search Tree and heap to store the fitness of the rules generated during the evolution [3], [26], [27]. These GAs remember all the candidate solutions generated and their respective fitness by organizing them into some efficient data structure. For example, a heap is one of the efficient data structures where insertion and deletion of an element take O $(\log_2 n)$ time where n is number of elements in the heap. In [3] authors have used a heap to organize the candidate solutions (Classification rules in If-Then form) and their fitness. During evolution whenever a rule is generated, the heap is searched for the rule. If the rule is found, its fitness is not re-evaluated but retrieved from the heap. If search in the heap fails, then the fitness of the rule is evaluated and inserted in the heap. This approach is effective to reduce the significant number of fitness evaluations, particularly towards the end of the GA when it is about to converge. However, such approaches need extra amount of memory and some additional computation effort. A novel research direction is to extract all the information required to compute the fitness of a classification rule and to store this information in a data structure. Later, the fitness of a classification rule can be computed from this information without any database scan.

D. Random Initialization of GA Population

If population of a genetic algorithm is initialized randomly then it produces individual rules that cover very few training instances. Therefore, initial rules in the population have zero or extremely low fitness and make selection procedure a random walk. In such situations, a GA either gives a poor set of rules or needs to run for thousands of generations for discovering better rules. To solve this problem researchers have proposed methods to bias the rule initialization process so that an initial population generates at least some individuals with non-zero fitness. One solution is seeding the initial population i.e. choose training instance as a seed for rule invention [3]. This seeded rule can be generalized to cover more number of instances from a dataset. In [33] initial population is created methodically using generalized uniform population method. In this approach, it is assumed that the lower and upper bound of genes in the chromosomes has been known. Firstly, two chromosomes are generated such that all genes of one represent lower bound and the other represent upper bound (complement of first one) and the other chromosomes are generated based on the complementation. Another solution is to use an entropy (or any other similar measure) based filter to bias an initial population to have more relevant attributes with high predictive power. Entropy is a measure of uncertainty and attributes with low entropy provide more information for prediction [3], [20]. It ensures that the relevant attributes are initialized with higher probability and GA starts with better rules, covering many training instances. This scheme increases initial population fitness and a better-fit initial population lead to the discovery of better-fit rules in lesser number of generations. Recently, some bottoms up approaches for seeding GA population have been applied. These approaches take a sample of training examples, generalize these examples to classification rules by keeping the relevant attributes in the rules and making rest of the attributes as ‘don’t care states’ [3], [20]. More research is required to seed the population of GAs following Pittsburgh approach.

E. Low Quality Dataset

An intelligent classification algorithm will not succeed to discover high-quality knowledge if it is applied to a low quality data [17]. Hence, it is essential to pre-process the data to improve quality of the dataset. One of the pre-processing tasks is feature selection. Feature selection method selects a subset of available features for constructing the model. Many attributes are either partly or completely irrelevant or redundant to the target concept. An unrelated attribute does not make any effect, and a redundant feature does not add anything new to the target concept. If the size of a dataset is large, learning takes a lot of time without removing these unwanted features. The goal of feature selections to decide on a small subset such that the resulting distribution of class is similar to the class

DOI 10.17148/IJARCCE.2016.56110

distribution given all feature values[21]. Removing of irrelevant and redundant features significantly reduces the running time of learning. In addition, as a form of data preparation task, continuous attribute can be discretized and these discretized attributes can be treated as categorical attributes [17].A pre-processing framework needs to be worked out for making the data clean and ready for discovering classification rules using GAs.

F. Niching Speciation

Simple Genetic Algorithm (SGA) converges to a single peak (though there are other peaks of equal fitness present in the search space) due to genetic drift [23]. Therefore, it is required to promote speciation, and hence diversity within the population. This goal is accomplished by promoting mating and replacement within the members of the same niche at the same time allowing some competition among the different niches. There have been various proposals for techniques and methodologies that encourage speciation within a GA population. In Cavicchio's (1970) pre-selection, an offspring (having better fitness than the parents) replace the most similar parent [29]. De Jong's crowding (1975) is a generalization of pre-selection, and uses a replacement policy called worst among most similar (WAMS) [29]. Goldberg and Richardson (1987) used the sharing concept to reduce the selection pressure to produce the fitness proportionate rule [23].In fitness sharing, this is accomplished by reducing the fitness of an individual proportionate to the number of individuals in the immediate neighbourhood, defined by some threshold distance. This reduces the height of the populated peaks in order that the individuals in each of the different peaks possess the same fitness. Thus, they all have equal probability to be selected in the subsequent generation, and diversity is maintained. In the context of rule discovery, if there are several individuals in the population correspond to the identical rule, the fitness of those individuals will be noticeably demoted, and so they will have lower probability of being selected [22].

Several enhancements to the original fitness-sharing algorithm are described in [23], [30]. A niche-size parameter (n*) is used to limit the maximum number of individuals in each niche. In the Sequential Niche Technique, run the GA repeatedly on the same problem. After each run, the optimized function is adjusted according to the location of solutions discovered in former iterations so that the optimum just found will not be located again. In [30], a clearing approach is introduced. In this procedure, subpopulations are determined in accordance with certain similarity measures. Those subpopulations are then cleared to allow evolution of other optima. Another appealing technique is K means clustering algorithm that is used to divide the population into k clusters of individuals, corresponding to k niches. The raw fitness of the individual is divided by the niche count of the cluster to which the individual belongs but this method entails some prior knowledge of the fitness function. Another method that can be used to foster GA to discover a diverse set of rules is using sequential coverage approach (also called as separate-and-conquer approach) [22]. The basic idea is that in the first run, the GA population is initialized using the complete training set and an empty model (i.e., set of rules) is created. After each run of the GA, the best rule evolved by the GA is added to the model and the instances covered by that rule are removed from the training set, so that the next run of the GA will generate a rule using smaller training set. This process continues until all examples of training set have been covered. This approach takes longer time because a GA is run to find individual rules and it ignores the role of rule interactions. The techniques like fitness sharing and crowding are customized for binary encoding. Most often, alpha numeric and other high level encoding schemes are used for discovery of classification rules. Devising new techniques for niche and speciation, specially tailored for discovery of classification rules awaits researchers ‘attention.

IV. CONCLUSION

This paper highlights the issues and challenges of applying GA based approaches for discovery of classification rules. It summarizes the manner in which the state-of-the-art research has addressed the issues like setting of GA parameters, seeding the population, niching and speciation, local convergence and computationally expensive fitness computations etc. The paper also points to the research directions to enhance the efficacy and efficiency of GAs in the domain of classification rule discovery. Research in these directions will make it viable to apply GAs for discovering classification models for larger datasets and in real time settings.

REFERENCES

[1] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 2nd ed., Morgan Kaufmann, 2006.

[2] A. A. Freitas, “A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery,” in Advances in Evolutionary Computing Theory and Applications, Springer-Verlag, NewYork, 2003, pp. 819-845.

[3] Saroj, Kapila, D. Kumar, and Kanika, “A Genetic Algorithm with Entropy Based Probabilistic Initialization and Memory for Automated Rule Mining,” in Proceedings of First International Conference on Computer Science and Information Technology, CCSIT, Bangalore, India, Springer, 2011, pp. 604–613.

[4] J. Vashishtha, D. Kumar, S. Ratnoo, and K. Kundu, “Mining Comprehensible and Interesting Rules: A Genetic Algorithm Approach,” International Journal of Computer Applications, vol. 31, no. 1, pp. 39–47, Oct. 2011.

[5] S. Dehuri and A. Ghosh, “Revisiting evolutionary algorithms in feature selection and nonfuzzy/fuzzy rule based classification,” WIREs Data Mining Knowledge Discovery, vol. 3, no. 2, pp. 83–108, Mar. 2013.

[6] C. Zhou, W. Xiao, T. M. Tirpak, and P. C. Nelson, “Evolving accurate and compact classification rules with gene expression programming,” IEEE Transactions on Evolutionary Computation, vol. 7, no. 6, pp. 519–531, Dec. 2003.

[7] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer Berlin Heidelberg, 1996.

[8] D.E. Goldberg: “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Publishing Company, Inc. MA, New York, 1989.

[9] B. M.Al-Maqaleh and H. Shahbazkia, “A Genetic Algorithm for Discovering Classification Rules in Data Mining,” International Journal of Computer Applications, vol. 41, no. 18, pp. 40–44, Mar. 2012.

DOI 10.17148/IJARCCE.2016.56110

[10] B. K. Sarkar, S. S. Sana, and K. Chaudhuri, “A genetic algorithm-based rule extraction system,” Applied Soft Computing, vol. 12, no. 1, pp. 238–254, Jan. 2012.

[11] J. Juan Liu and J. T.-Y. Kwok, “An extended genetic rule induction algorithm,” in Proceedings of the Congress on Evolutionary Computation, IEEE Press, LaJolla, 2011, vol. 1, pp. 458–463.

[12] M. V. Fidelis, H. S. Lopes, and A. A. Freitas, “Discovering comprehensible classification rules with a genetic algorithm,” in Proceedings of the Congress on Evolutionary Computation, La Jolla, CA, USA, IEEE, 2000, vol. 1, pp. 805–810.

[13] E. Noda, A. A. Freitas, and H. S. Lopes, “Discovering interesting prediction rules with a genetic algorithm,” in Proceedings of the Congress on Evolutionary Computation(CEC-99), Washington D.C., USA,1999, vol. 2, pp.1322- 1329.

[14] J. Brest, B. Bošković, S. Greiner, V. Žumer, and M. S. Maučec, “Performance comparison of self-adaptive and adaptive differential evolution algorithms,” Soft Computing, vol. 11, no. 7, pp. 617–629, Jul. 2006.

[15] P. Sharma and S. Ratnoo, “A review on Discovery of Classification Rules using Pittsburgh Approach,” International Journal of Artificial Intelligence and Knowledge Discovery, vol. 4, no. 3, pp. 6–16, Jul. 2014.

[16] P. Sharma and S. Ratnoo, “Bottom-up Pittsburgh approach for discovery of classification rules,” in International Conference on Contemporary Computing and Informatics (IC3I), Mysore, 2014, pp. 31–37.

[17] A. A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms. Natural Computing Series. Springer, New York ,2002.

[18] S. Saroj and K. K. Bharadwaj, “A Parallel Genetic Algorithm Approach for Automated Discovery of Censored Production Rules,” in Proceedings of International Conference on Artificial Intelligence and Application (IASTED), ACTA Press, Innsbruck, Austria, 2007, pp. 435-441.

[19] M. L. F. Herrera, “Hierarchical Distributed Genetic Algorithms,” International Journal of Intelligent Systems, vol. 14, no. 11, pp. 1099–1121, 1999.

[20] Kapila, Saroj, Dinesh Kumar, Kanika, “A Genetic Algorithm with entropy based initial bias for automated rule mining” in Proceedings of IEEE Conference on Computer and Communication Technology (ICCCT), MNNIT Allahabad, 2010, pp 491-495.

[21] M. Dash and H. Liu, “Feature selection for classification,” Intelligent Data Analysis: An International Journal, vol. 3, pp. 131–156, 1997.

[22] O. Maimon and L. Rokach, “Introduction to Knowledge Discovery and Data Mining,” in Data Mining and Knowledge Discovery Handbook, Eds. Springer US, pp. 1–15, 2009.

[23] J. Gan and K. Warwick, “A Variable Radius Niching Technique for Speciation in Genetic Algorithms,” in Proceedings of the 2nd Annual Conference on Genetic and Evolutionary Computation, San Francisco, CA, USA, 2000, pp. 96–103.

[24] Welcome to IASRI - a premier institute of Statistics and Computing in the field of Agriculture. (n.d.). Indian Council of Agricultural Research.[Online]Available:http://iasri.res.in/ebook/win_school_aa/notes/ Genetic_Algorithm .pdf . Accessed May 19, 2016.

[25] A. E. Eiben, R. Hinterding, and Z. Michalewicz, “Parameter control in evolutionary algorithms,” IEEE Transactions on Evolutionary Computation, vol. 3, no. 2, pp. 124–141, Jul. 1999.

[26] V. B. Gantovnik, C. M. Anderson-Cook, Z. Gürdal, and L. T. Watson, “A genetic algorithm with memory for mixed discrete–continuous design optimization,” Computers & Structures, vol. 81, no. 20, pp. 2003–2009, Aug. 2003.

[27] S. Louis and G. Li, “Augmenting Genetic Algorithms with Memory to Solve Traveling Salesman Problems,” in Proceedings of the Joint Conference on Information Sciences, Duke University Press, 1997, pp. 108–111.

[28] P. Sharma and Saroj, “Discovery of Classification Rules Using Distributed Genetic Algorithm , ” Procedia Computer Science, vol. 46, pp. 276–284, 2015.

[29] W. Cedeño and V. R. Vemuri, “Analysis of speciation and niching in the multi-niche crowding GA,” Theoretical Computer Science, vol. 229, no. 1–2, pp. 177–197, Nov. 1999.

[30] M. B. Fayek, N. M. Darwish, and M. M. Ali, “Context based clearing procedure: A niching method for genetic algorithms,” Journal of Advanced Research, vol. 1, no. 4, pp. 301–307, Oct. 2010.

[31] S.-T.-J. Salma-Tuz-Jakirin, A. Ahmed Ferdaus, and M. Afrin Khan, “A Genetic Algorithm Approach using Improved Fitness Function for Classification Rule Mining,” International Journal of Computer Applications, vol. 97, no. 23, pp. 12–18, Jul. 2014.

[32] X. J. Shi and H. Lei, “A Genetic Algorithm-Based Approach for Classification Rule Discovery,” in 2008 International Conference on Information Management, Innovation Management and Industrial Engineering, Taipei, 2008, vol. 1, pp. 175–178.

[33] K. K. Gündoğan, B. Alataş, and A. Karci, “Mining Classification Rules by Using Genetic Algorithms with Non-random Initial Population and Uniform Operator,” Turkish Journal of Electrical Engineering & Computer Sciences ,vol. 12, no. 1, pp. 43–52, Jun. 2004.

[34] D. L. A. de Araujo, H. S. Lopes, and A. A. Freitas, “A parallel genetic algorithm for rule discovery in large databases,” in IEEE International Conference on Systems, Man, and Cybernetics,Tokyo, vol. 3, pp. 940–945, 1999.

[35] M. Srinivas and L. M. Patnaik, “Adaptive probabilities of crossover and mutation in genetic algorithms,” IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 4, pp. 656–667, Apr. 1994.

[36] K. G. Srinivasa, K. R. Venugopal, and L. M. Patnaik, “A self-adaptive migration model genetic algorithm for data mining applications,” Information Sciences, vol. 177, no. 20, pp. 4295–4313, Oct. 2007.

[37] E. Pellerin, L. Pigeon, and S. Delisle, “Self-adaptive parameters in genetic algorithms,” presented at the Data Mining and Knowledge Discovery: Theory, Tools, and Technology VI, Belur V. Dasarathy Orlando, FL, 2004, vol. 5433, pp. 53–64.

DOI 10.17148/IJARCCE.2016.56110