Analysis of Nine Instance-Based Genetic Algorithm Classifiers Using Small Datasets

Hossin, M., Mahudin, F., Din, I and Mat, A.R
Faculty of Computer Science and Information Technology,
Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia.
hmohamma@unimas.my

***Abstract*—The application of genetic algorithm (GA) has emerged covering various areas including data classification. In data classification, most studies of GA were focused on the enhancement of GA and development of different types of GA classifiers. To the best of our knowledge, there is no study has been conducted to examine the influence of GA operators based on the size of data set towards training time and generalization ability. Therefore, this study develops and compares nine Instance-based genetic algorithm (IbGA) classifiers with different combinations of GA operators. The goal of this comparison is to examine and identify the best combination of GA operators which have performed better on generalization ability and training time efficiency. Nineteen benchmark data sets were used in this study. The non-parametric statistical tests were applied to justify the comparison results. The statistical tests suggest that the combination of roulette wheel selection and uniform crossover operator is the best combination of IbGA model although the training time is a bit lengthier than compared to other IbGA models..**

***Index Terms*—Data Classification; Genetic Algorithm; Instance-Based Classifier.**

I. Introduction

Typically, genetic algorithm (GA) was used as an optimizer to solve complex problems. Since its inception, the use of GA has been expanded to solve data classification. There are two types of GA classifiers; rule-based GA (RbGA) and instance-based GA (IbGA) [1]. The IbGA classifier was inspired from the drawback of the nearest neighbor (NN) algorithm. The large storage of prototypes and long response time classification are two major drawbacks of NN classifier. Due to these disadvantages, IbGA was proposed to reduce the number of prototypes as much as possible while preserving the NN classifier performance. On the other hands, the RbGA classifier was inspired from the rule-based approach. In RbGA, each chromosome is represented by different rules that generated randomly. Each allele (or known as gene) represents each data attribute and represented by binary string (0 and 1) based on the possible values for each attribute. Normally, each allele has different length of binary bits. Then, the rule is generated by employing information measure such like entropy [2], or ranking with correlation coefficients [3].

Interestingly, many studies were done on the RbGA as compared to IbGA. From the review, the RbGA has been applied to solve large data sets [4]. Meanwhile, IbGA seems less attractive to researchers due to its complicated framework representation and optimization process. In IbGA, the process of building the classifier is a stochastic process where the optimal n reference set is searched using optimization process. Due to optimization process, the finding process has become computational costly when large data is involved. Although IbGA is less attractive, the reported generalization performance of IbGA was superior or at par as compared to other instance-based classifiers or other types of classifiers for many benchmark data sets [5, 6].

In order to design the best classification algorithm, many studies focus on both data and algorithmic level had been conducted. For algorithmic level, it includes the advanced design of algorithm and improvement in order to get better results for a specific domain. In contrast, this study attempts to analyse the algorithmic level of operators used in GA towards training time and generalisation ability. According to Abdoun and Abouchabaka [7], this analysis is important because the performance of GA is totally dependent on the selection of appropriate genetic operators. To the best of our knowledge, there are no known studies that focus on the effect of different combination GA operators towards generalization ability and training time efficiency. However, Andrade et al., [8] did conduct a comprehensive analysis to examine the effect of GA operator combinations in route searching problem in IP network domain. They noticed that each combination of GA operators did influence the performance of GA in routing searching problem. They also conclude that Stochastic Random Sampling (SRS selection) and uniform crossover combination was able to achieve less processing time as compared to other GA operator combinations. Thus, it is important to investigate the influence of each GA operator combinations towards the performance of IbGA in terms of training time and generalization ability. Through this study, two research questions have been identified:

- How different combinations of GA operators influence the performance of the IbGA towards training time efficiency and generalization ability based on several benchmark data sets?
- What is the best combination of GA operators that give better generalization performance and produce less training time based on several benchmark data sets?

The scope of this study is confined to the modification of classical IbGA classifier. This study will examine the influence of different combinations GA operators towards training time efficiency and generalization ability. Three selection techniques, three crossover operators, and one mutation technique were used for this particular study. In total, nine various IbGA classifiers were developed. This study employs a standard accuracy measure and training time (in second) to measure the performance of each proposed IbGA classifier on various benchmark data sets. 19 benchmark data sets which are binary-class datasets were employed for both training and testing process.

II. Materials And Methods

The idea of using genetic algorithm (GA) as a classifier for data classification was initiated by Kuncheva and Bezdek [5]. They developed and conducted an experimental study to compare between two different types of prototype classifiers which are random search and genetic algorithm. Surprisingly, the result was promising for GA as compared to random search prototype based classifier. Similar to the traditional GA, the IbGA also conserved the five principal procedures of GA. These procedures are population initialization, mating strategy, crossover, mutation and replacement strategy.

In IbGA, the chromosome representation was similar to traditional GA where the chromosome is coded using binary string {0 or 1}. The only difference of this coded is the denotation of each gene. In IbGA, each gene denotes each instance in the training dataset. The string 1 denotes the gene or the particular instance has become a representative prototype for a certain class and 0 is otherwise. Those selected instances are known as reference set (cardinality) which is selected randomly. Through selection strategy, the reference set will be chosen from the existing instances in training dataset D by limiting the numbers. Let's assume that each chromosome C_{all} be the set of given j instances $C_{all} = \{x_1, x_2, \ldots, x_j\}$ where $j = 1,2, \ldots, j$ are data from c classes in a particular data set D. Let C be the selected data points and a reference set (solution) $C \in C_{all}$. Every C is coded using binary string {0 or 1} based on the length j and represented as 0 for inactive representative and 1 as an active representative. This representation can be illustrated as depicted in Figure 1.

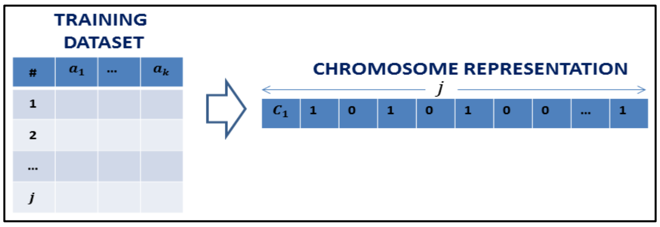

Figure 1: IbGA Chromosome Representation

Through this chromosome representation, the initial population generation of chromosomes can be initialized. In Kuncheva and Bezdek [5], the implementation of the other GA operators was implemented as stated in Table 1.

Table 1
Summary of IbGA Operators

GA operators	IbGA
Mating strategy	Roulette wheel technique
Crossover	Uniform crossover
Mutation	$P_m = 0.01$
Replacement strategy	Elitist approach

For fitness function, the IbGA employed a combination of classification accuracy and penalty function. This fitness function will drive the IbGA model to obtain minimal reference set (cardinality) and simultaneously aim for highest classification accuracy. This fitness function can be written as to find a set of C-prototypes C^* that satisfies $C^* = \arg\max_{Scx} F(C)$, where $F(B)$ is the objective function. In this case, the F comprises two components.

$$F(B) = A(B) - \propto f(|B|)$$

The first component $A(C)$ denotes the classification accuracy when using C as the reference set. Meanwhile, the second component $\propto f(|C|)$ denotes the function of cardinality (number of prototypes) of C weighted by the coefficient $\propto > 0$. Since the objective function is to obtain minimal cardinality, the parameter T has been proposed. In this case, T is predefined value where it can force the IbGA to converge to predefined number of prototypes $|C^*| = T$.

A. *Nine Instance-based GA classifiers*

This study proposes and develops nine IbGA models that inspired from classical GA classifier [5]. Three popular selection operators (mating strategy) and three crossover operators were selected for comparison. However, we employed only one mutation technique which was used in the classical GA classifier. Table 2 briefly describes the nine IbGA models that derived from the different combination of selection and crossover operators. The abbreviations were used for analysis and comparison purposes. All IbGA models that constructed in this study were implemented using MATLAB Script 2009(b).

Table 2
Nine Various IbGA Models for Analysis

No.	Proposed Instance-based GA (IGA) Classifiers	Abbreviation
1.	IGA with random selection and one-point crossover	RS1
2.	IGA with random selection and two-point crossover	RS2
3.	IGA with random selection and uniform crossover	RSU
4.	IGA with tournament selection and one-point crossover	TS1
5.	IGA with tournament selection and two-point crossover	TS2
6.	IGA with tournament selection and uniform crossover	TSU
7.	IGA with roulette wheel selection and one-point crossover	RW1
8.	IGA with roulette wheel selection and two-point crossover	RW2
9.	IGA with roulette wheel selection and uniform crossover	RWU

B. *Experimental Setup*

For the experiment purposes, the basic parameters were setup based on classical IbGA [5]. The only different parameter is P_{ini} due to the different number of volume of benchmark data sets were used.

Population size: $T_{pop} = 20$
Initial probability search: $P_{ini} = 0.05$, restricted GA
Total number of generation: $M_{gen} = 500$
Mutation rate: $P_m = 0.025$
Weighting coefficient for penalty term: $\alpha = 0.1$

Predefined number of prototype: $T = 15$

For tournament selection, the tournament size, $T_s = 3$ was used for the entire experiments.

This study employs two performance measures to compare the nine IbGA models. The accuracy measure was used to measure the generalization ability of testing data sets. All data sets were divided into 10-folds using k-fold cross validation technique. Average testing accuracy for each data set was used for evaluation comparison. In addition, the time taken (in second) during the training process was used for measuring the speed of each IbGA model in completing the training process. There are 19 binary data sets were used for the evaluation and analysis purposes as shown in Table 3. These data sets represent the classical benchmark data sets which vary in terms of the number of volume and attributes. All these data sets were categorized as small data sets.

For a better comparison analysis, the statistical analysis was employed in this study to compare each IbGA model against other IbGA model [9]. The *ss* denotes win/draw/loss, *pw* denotes Wilcoxon sign-rank test significant value, and *ps* denotes Sign test significant value. The significant difference between the two models was inferred through *p*-value. If the *p*-value is below than 0.05 (5%), then the two observed IbGA models have a significant difference. All experiments were run using one personal computer (Acer, Aspire V3-471G,Intel(R) core(TM) i5-3210M CPU @2.50GHz, 6.00 GB RAM, 64-bit OS) to avoid timing bias. The Statistical Package for Social Sciences version 22.0 (SPSS 22.0) was used for analysing the statistical results.

Table 3
Brief Description of Benchmark Datasets (Binary-Class)

Data Set	Abbrevations	NoI	Size	NoA	Dimension
Hepatitis Domain	Hepa	155	S	19	2945
Parkinsons disease	Parkinson	195	S	22	4290
Wisconsin Prognostic Breast Cancer	BCP	198	S	33	6534
Connectionist Bench (Sonar, Mines vs. Rocks)	Sonar	208	S	60	12480
SPECT heart data	SPECT	267	S	22	5874
SPECTF heart data	SPECTF	267	S	44	11748
Statlog Heart Disease	Heart270	270	S	13	3510
Haberman’s Survival data	Haber1	306	S	3	918
Bupa Liver Disorders	Liver	345	S	6	2070
Johns Hopkins University Ionosphere data set	Ionos	351	S	34	11934
1984 U.S Congressional Voting Records	Votes	435	S	16	6960
Musk Clean1 data	Musk11	476	S	166	79016
3 consecutive bits of 9 features are true	Three9	512	S	9	4608
Wisconsin Diagnostic Breast Cancer	BCD	569	S	30	17070
Statlog Australian Credit approval	CardAus	690	S	14	9660
Wisconsin Original Breast Cance	BCO	699	S	9	6291
Blood Transfusion Service Center	Trans1	748	S	4	2992
Pima Indian Diabetes	Pima	768	S	8	6144
Statlog German Credit	CardGer	1000	S	24	24000

Note: NoI-no. of instances, NoA-no. of attributes, S-small dataset

III. Results and Discussions

The average accuracy measure was used to discriminate the best models. The time taken from training process also been recorded in order to identify the time efficiency of each IbGA model. This time taken will be used to support the assumption made for the selected best model. The discussions of previous research are also discussed consolidating the obtained results in this study.

A. *Generalisation Performance Analysis*

Table 3 shows the average testing accuracy of all data sets for all IbGA models. In terms of generalisation ability, there is no absolute model can be claimed as the best model for IbGA classifier. Noticeably, only the RWU model won six over 19 data sets as compared to other IbGA models. However, the obtained results show no large differences among all nine IbGA models. Thus, it is difficult to suggest the best combination of GA operators. To examine the obtained results further, the statistical analysis of descriptive analysis (ss), sign test (ps) and Wilcoxon test (pw) were used and the analysis results are depicted in Table 4.

Through the statistical analysis, Table 4 shows that for the random search group, the RS1 is able to outperform both RS2 and RSU. Although RS1 is able to outperform those two models, the differences are not significant for both sign and Wilcoxon test. The similar pattern can be seen for the tournament selection group. Although TS2 was able to outperform TS1 and TSU, that achieving does not significant for both sign and Wilcoxon tests. On the contrary, for the roulette wheel group, RWU was shown able to outperform other eight models. However, RWU only significantly outperformed against RS1, RS2, RSU, TS1, TS2, TSU, and RW1 respectively. The performance of RWU was significant for both sign and Wilcoxon test. From the result in Table 4, this study could suggest that RWU was the best combination of selection and crossover operators for IbGA classifier.

B. *Time Efficiency*

Table 5 demonstrates the time taken for nine IbGA models to complete the training process. The aim is to find the IbGA model that requires the least time to complete the training process. Generally, by referring to the average time taken obtained by each model, the IbGA model that employed random selection and tournament selection with one or two-point crossover was timely efficient. Not surprisingly, any combination of uniform crossover took a longer time to complete the training process. All roulette wheel combinations also show longer time to complete the training process due to its complexity process.

From this experiment, we cannot simply conclude on the best combination of GA operator that can produce a better result in terms of generalisation ability and training time efficiency simultaneously. However, when we observe carefully the training time taken for each IbGA model, all results were almost similar and the differences for each data set are too small if we do a one-to-one comparison. Therefore, through both observations on accuracy and time taken, we can generally conclude that the combination of the Roulette Wheel and uniform crossover is the best combination, although the training time is a bit longer than the other IbGA models.

Table 3
Overall Results for the Average Testing Accuracy of Nine IbGA Models

Data sets	Average testing accuracy								
	RS1	RS2	RSU	TS1	TS2	TSU	RW1	RW2	RWU
LIVER	46.37	46.64	47.23	47.19	52.40	44.08	45.19	51.58	53.09
BCD	86.65	85.25	82.79	83.14	88.40	84.54	82.41	85.95	87.87
BCO	94.85	92.99	92.57	90.60	93.71	92.42	93.26	92.4	93.28
BCP	56.45	36.87	50.97	51.95	49.95	48.92	47.95	56.58	52.95
CARDAUS	71.74	73.33	72.47	77.25	73.04	76.09	71.74	75.65	73.92
CARDGER	59.20	57.70	60.4	60.70	58.40	56.90	58.10	59.10	62.40
HABER1	47.97	45.10	47.08	42.17	47.02	46.69	49.72	49.33	53.60
HEART270	60.00	62.22	65.56	66.67	60.74	55.59	64.82	61.48	65.93
HEPA	72.92	64.62	57.17	60.13	64.63	57.21	64.42	77.42	69.79
IONOS	72.65	68.68	72.08	70.64	68.05	67.50	68.08	68.94	73.76
MUSK11	54.44	57.23	58.42	57.40	53.83	56.29	55.09	56.71	56.75
PARKINSON	72.84	73.82	72.21	74.26	70.24	75.82	69.76	73.82	75.90
PIMA	62.99	63.41	66.80	63.02	61.34	65.00	67.31	62.75	67.71
SONAR	49.50	51.95	51.40	51.93	53.43	49.98	51.93	52.88	51.38
SPECT	60.61	64.30	62.44	56.01	57.66	59.52	62.51	67.45	62.57
SPECTF	56.21	61.42	60.71	58.75	56.21	61.81	65.31	59.71	64.84
THREE9	63.29	62.70	61.33	62.12	62.66	63.87	61.14	63.07	60.55
TRANS1	70.32	66.31	67.92	66.57	68.84	69.27	70.06	69.52	70.07
VOTES	81.10	80.88	82.51	82.34	84.32	81.60	77.90	82.06	79.74
Average	65.27	63.97	64.85	64.36	64.47	63.85	64.56	66.65	67.16

Note: Bold and underline – the best performances

Table 4
Comparison of Different IbGA Models in term of Average Testing Accuracy

Evaluation measure	GA models (row)	Analysis (test)	GA models (column)								
			RS1	RS2	RSU	TS1	TS2	TSU	RW1	RW2	RWU
Average accuracy	RS1	ss		10/0/9	10/0/9	9/0/10	12/1/6	11/0/8	11/1/7	7/0/12	6/0/13
		ps		1.000	1.000	1.000	0.238	0.648	0.481	0.359	0.167
		pw		0.601	0.872	0.520	0.327	0.227	0.500	**0.049**	**0.049**
	RS2	ss			9/0/10	9/0/10	9/0/10	10/0/9	11/0/8	5/1/13	5/0/14
		ps			1.000	1.000	1.000	1.000	0.648	0.096	**0.064**
		pw			0.305	0.936	0.794	0.872	0.841	**0.016**	**0.004**
	RSU	ss				10/0/9	10/0/9	12/0/7	11/0/8	8/0/11	4/0/15
		ps				1.000	1.000	0.359	0.648	0.648	0.019
		pw				0.546	0.717	0.227	0.421	0.136	**0.006**
	TS1	ss					9/0/10	10/0/9	11/1/7	8/0/11	6/0/13
		ps					1.000	1.000	0.481	0.648	0.167
		pw					1.000	0.778	0.811	0.091	**0.009**
	TS2	ss						11/0/8	11/0/8	5/0/14	5/0/14
		ps						0.648	0.648	0.064	0.064
		pw						0.573	0.748	**0.022**	**0.006**
	TSU	ss							7/0/12	6/0/13	3/0/16
		ps							0.359	0.167	**0.004**
		pw							0.507	**0.033**	**0.004**
	RW1	ss								6/0/13	3/0/16
		ps								0.167	**0.040**
		pw								**0.046**	**0.020**
	RW2	ss									8/0/11
		ps									0.359
		pw									0.445

Note: Bold and underline – significance value, ss-descriptive analysis (win/draw/loss), ps-sign test, pw-Wilcoxon test

Table 5
Overall Results for the Average Training Time of Nine GA Models

Data sets	Average training time (in second)								
	RS1	RS2	RSU	TS1	TS2	TSU	RW1	RW2	RWU
LIVER	062.4	062.0	063.0	065.3	065.1	065.0	065.6	066.4	065.2
BCD	330.9	320.8	322.4	316.6	309.1	322.6	330.0	327.7	327.4
BCO	264.7	261.7	267.5	257.2	252.5	257.5	266.3	265.3	261.1
BCP	046.6	048.4	049.4	052.7	052.9	052.8	050.4	049.8	049.6
CARDAUS	307.7	320.6	309.6	296.6	296.8	291.3	302.7	312.2	302.0
CARDGER	849.2	856.4	867.7	819.8	825.2	815.2	955.5	957.2	963.2
HABER1	045.3	045.2	045.7	046.5	047.7	047.3	046.8	047.1	046.8
HEART270	053.8	051.9	053.4	056.7	057.4	057.0	054.9	055.0	054.0
HEPA	024.1	024.1	024.2	026.2	026.8	026.5	025.3	025.6	024.9
IONOS	141.5	141.7	142.0	139.0	139.5	143.3	144.6	147.4	143.5

Data sets	Average training time (in second)								
	RS1	RS2	RSU	TS1	TS2	TSU	RW1	RW2	RWU
MUSK11	925.4	948.5	916.8	**<u>833.1</u>**	860.2	841.9	914.3	936.9	943.3
PARKINSON	**<u>037.3</u>**	037.8	037.5	040.9	040.9	041.3	039.6	039.5	039.2
PIMA	299.3	312.3	305.6	291.6	**<u>288.9</u>**	293.8	296.3	293.5	296.8
SONAR	**<u>077.9</u>**	078.3	078.1	086.5	088.2	091.9	080.5	081.9	079.6
SPECT	**<u>065.8</u>**	066.3	066.4	070.0	069.5	071.0	069.4	070.5	069.6
SPECTF	**<u>098.9</u>**	099.5	099.7	103.6	105.6	106.5	102.7	102.9	101.4
THREE9	148.3	147.7	151.7	**<u>141.9</u>**	143.2	145.2	152.7	150.9	150.7
TRANS1	241.2	240.5	243.2	**<u>232.8</u>**	235.0	235.7	240.1	239.5	238.0
VOTES	141.4	139.8	138.6	**<u>137.5</u>**	138.2	141.6	143.6	142.8	143.5
Average	219.0	221.2	220.1	**<u>211.3</u>**	212.7	213.0	225.3	227.0	226.3

Note: Bold and underline – the best performances

IV. Discussion

As mentioned above, the result of this study suggests that the roulette wheel and uniform crossover are the best combination GA operators for IbGA. Surprisingly, this combination was also being used by Kuncheva and Bezdek [5]. However, this finding is contradicted to Magalhães-Mendes [10] study on a different domain. This study concludes that the combination of the roulette wheel selection and the one-point crossover is the best combination for scheduling problems. Based on this contradiction finding, it is clearly shown that a different domain requires a different type of GA operator combination. For training time efficiency, the combination of the roulette wheel and uniform crossover shows the satisfactory performance. This result was similar to the study done by Andrade et al. [8]. As shown in Table 5, obviously the combination of tournament selection and one-point crossover shows the least processing time for building the IbGA classifier. This finding shows a similar conclusion made by Zhong et al. [11] in a different domain of study. We also found that all IbGA models require more training time when the size of data set is increased. For the biggest data set (German Credit Card), all models took up to 14 to 16 minutes to complete the training process. Therefore, it shows clearly that the IbGA is a computational costly algorithm for large data set.

V. Conclusion

This paper proves that each combination of GA operators did influence the generalization ability and the training time efficiency of IbGA models. Through the experiments conducted, this study concludes that the best combination of GA operators in term of testing accuracy was RWU model of IbGA (Roulette wheel and uniform crossover) and the least processing time was TS1 model of IbGA (tournament selection and one-point crossover). In addition, we also conclude that all nine IbGA models are computational costly algorithm. We believe that this computational cost is caused by its chromosome coded. We found that the bigger data that we have, the longer chromosome was created. With longer chromosome coded, it will automatically affect the other GA procedures, especially the evaluation of fitness function of each chromosome in generation population and mutation process. Therefore, the GA chromosome coded is needed to be improved.

Acknowledgments

This research was funded by the Universiti Malaysia Sarawak under grant F08/SpSTG/1368/16/10

References

[1] S. Garcia, J. Derrac, J. R. Cano, and F. Herrera, "Prototype Selection for Nearest Neighbor Classification: Taxonomy and Empirical Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 417-435, 2012.

[2] L. Yang, D. H. Widyantoro, T. Ioerger, and J. Yen, "An entropy-based adaptive genetic algorithm for learning classification rules," In *Evolutionary Computation, 2001, Proceedings of the 2001 Congress on*, vol. 2, pp. 790-796, 2001. IEEE

[3] J. Liu, H. Iba, and M. Ishizuka, "Selecting informative genes with parallel genetic algorithms in tissue classification," *Genome Informatics*, vol. 12, pp. 14-23, 2001.

[4] P. Vivekanandan, M. Rajalakshmi, and R. Nedunchezhian, "An intelligent genetic algorithm for mining classification rules in large datasets," *Computing and Informatics*, vol. 32, no. 1, pp. 1-22, 2013.

[5] L. I. Kuncheva and J. C. Bezdek, "Nearest prototype classification: clustering, genetic algorithms, or random search?," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 28, no. 1, pp. 160-164, 1998.

[6] H. Ishibuchi and T. Nakashima, "Multi-objective pattern and feature selection by a genetic algorithm," In *GECCO*, pp. 1069, 2000.

[7] O. Abdoun and J. Abouchabaka, "A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem," *arXiv preprint arXiv:1203.3097*, 2012.

[8] A. V. Andrade, L. D. Errico, A. L. L. D. Aquino, L. P. D. Assis, and C. H. N. D. R. Barbosa, "Analysis of selection and crossover methods used by genetic algorithm-based heuristic to solve the lsp allocation problem in mpls networks under capacity constraints," 2008.

[9] N. García-Pedrajas, J. A. R. Del Castillo, and D. Ortiz-Boyer, "A cooperative coevolutionary algorithm for instance selection for instance-based learning," *Machine Learning*, vol. 78, no. 3, pp. 381-420, 2010.

[10] J. Magalhães-Mendes, "A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem," *WSEAS transactions on computers*, vol. 12, no. 4, pp. 164-173, 2013.

[11] J. Zhong, X. Hu, M. Gu, and J. Zhang, "Comparison of performance between different selection strategies on simple genetic algorithms," In *Computational Intelligence for Modelling, Control and Automation, 2005 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on*, vol. 2, pp. 1115-1121, Nov. 2005. IEEE.