# Improved Genetic Algorithm using New Crossover Operator

Joe G. Lagarteja, Bobby D. Gerardo, and Ruji P. Medina

***Abstract*—**The complexity of existing crossover operators used in Genetic Algorithm is a critical factor that affects performance due to its negative impact on processing time. In this paper, a new crossover operator called Push and Pop Genes Exchange Operator (PPX) is introduced and its performance evaluated in terms of processing time. Results of comparative performance with six crossover operators show that PPX performed better in terms of processing time across various population size, with improvements ranging from 0.6% when compared to shuffle crossover at n=100 to 24.8% when compared to the half-uniform crossover operator at n=30. Results also show that PPX performed better with increase in population with a maximum of 13.1% when population was increased from 30 to 100. The results confirm that PPX improved the performance of Genetic Algorithm by reducing the complexity of crossover operation when compared to the existing operators.

***Index Terms*—** Push and Pop Genes Exchange Operator, Genetic Algorithm, Crossover operator.

## I. Introduction

Genetic Algorithm (GA) is a search-based optimization technique that uses the principles of genetics and natural selection to find optimal or near optimal solutions for difficult problems [1]. It is particularly useful for machine learning applications

An important innovation made to GA was the introduction of population-based algorithm with operators for crossover and mutation. The crossover operator is responsible for producing offsprings by way of recombining information from two parents, thus providing major exploratory mechanism of the algorithm [1]. On the other hand, mutation prevents convergence of the population by flipping a small number of randomly selected bits ensuring the continuous introduction of variation [1]. The unique cooperation between crossover and mutation, together with selection, provides the driving force behind GA [2]. The complexity of crossover operators play an important role in searching and providing solution to a problem since a more complex crossover operator results in longer processing times. This becomes even more pronounced when dealing with search problems of greater complexities.

This paper details the modification of GA utilizing a novel and less complex crossover operator called Push and Pop Exchange Genes Crossover (PPX). A comparison of its performance with those of existing crossover operators in terms of processing time using real data is then given to show how the use of simpler crossover operator is able to improve the performance of GA.

## II. Related Literature

Genetic algorithms have a recombination operation which seems to be closest to the natural paragon and crossover operator is used to mimic biological recombination between two single chromosome organisms [10]. Various crossover operators have been utilized but these have complexity issues that negatively affect the GA.

### *A. Segmented Crossover*

Segmented crossover represents a variant of N-point crossover. In this crossover, the number of crossover points is not constant at segment switch rates we used which specify the probability that a segment will end at any point in the string. Starting from first position in a string, one real-valued number $q$ and one natural number $j$ are generated. The number $q$ represents the probability that $j$ will be a crossover point. [3][5]

### *B. Shuffle Crossover*

Shuffle crossover is similar to one-point crossover. First, a single crossover position is selected. Before the variables are exchanged, they are randomly shuffled in both parents. After recombination, the variables in the offspring are un-shuffled in reverse. This removes positional bias as the variables are randomly reassigned each time crossover is performed. In a way, shuffle crossover is similar to uniform crossover but different in that uniform crossover exchanges bits and not segments like shuffle crossover. Furthermore, in uniform crossover bits exchanged follow a binary distribution and in shuffle crossover bits follow uniform distribution, as in single-point crossover. [2][3]

### *C. Reduced Surrogate Crossover*

To reduce the chance of producing clones Booker suggested examining the selected parents to define suitable crossover points. A reduced surrogate crossover operator reduces parent strings to a skeletal form in which only those bits that differ in two parents are represented. Recombination is then limited only to positions of bits in reduced surrogates. Single-point crossover was used for recombination of skeletal forms of parents.

Single-point crossover operator can produce parents' clones; to avoid that reduced surrogate crossover should be used. If at least one crossover point occurs between the first and last bits in reduced surrogate, then the offspring will never duplicate the parents. Also, reduced surrogate will cause that recombination process equally weights the probability of generating each offspring which can potentially be produced by an operator.

Single-point crossover in any continuous region of matching bits in parents produces same offspring, and thus introducing bias for some offspring. Reduced surrogate removes that kind of potential bias. [1][3][5]

### *D. Half-uniform Crossover*

The half uniform crossover schemes (HUX), exactly half of the nonmatching bits are swapped. Thus, first the Hamming

distance (the number of differing bits) is calculated. This number is divided by two. The resulting number is how many of the bits that do not match between the two parents will be swapped [9].

### *E. Uniform Crossover*

Single and multi-point crossover defines cross points as places between loci where an individual can be split. Uniform crossover generalizes this scheme to make every locus a potential crossover point. A crossover mask, the same length as the individual structure is created at random and the parity of the bits in the mask indicate which parent will supply the offspring with which bits. To avoid problems with genes locus, it is good to use uniform crossover.

Uniform crossover disrupts schema with great probability but searches larger problem space. For uniform crossover, the number of effective crossing points is not fixed, but will average to l/2 where l represents string length. [3][4][5]

### *F. Two-Point Crossover*

In two-point crossover, both parental genotypes are split at two points, constructing a new offspring by using parts number one and three from the first, and the middle part from the second ancestor.

When using two-point crossover we can expect poorer performance results because building blocks are more likely to be disrupted. From other point of view using two-point crossover will enable searching problem space more thoroughly. Using single-point and two-point crossover operator prevents schema to be disrupted, but when population becomes homogeneous, search space becomes smaller. [3][4]

## III. Modified Algorithm

A new PPX crossover operator was introduced into GA in order to speed up the crossover process and reduce processing time. PPX follows the concept of stacking using a Last In First Out (LIFO) approach. The proposed modified algorithm consists of five steps, as follows:

1. **Create** a *stack*
2. **Repeat**
   **Push** element into the stack
   **If** the stack is full
   *End*
   **End**
3. **Until** *loop=numElement*
4. **Repeat**
   **Pop** the element from the stack
5. **While** *stack<>empty*

In the initial step, an *n* number of elements (or parents) for the crossover process are selected. In the succeeding steps, a collection of elements are subjected to two principal operations: *push*, and *pop*, which remove the most recently added element that has not yet been removed. The order in which the elements come off the stack is indicative of its LIFO approach.

The restrictions that apply to the stacks are shown in Figures 1 and 2 showing the elements before and after the crossover process, respectively. For elements A, B, C, D and E added to the stack in that order, element E is first to be removed since it was the last element inserted into the stack in consonance with LIFO.

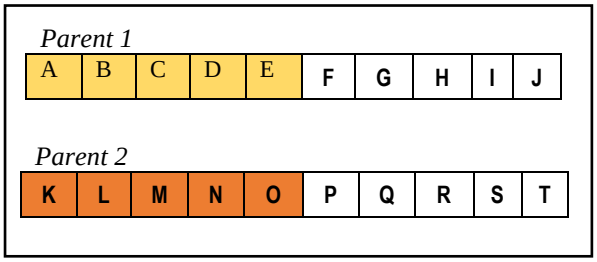


**Fig. 1**. Elements before the crossover process.

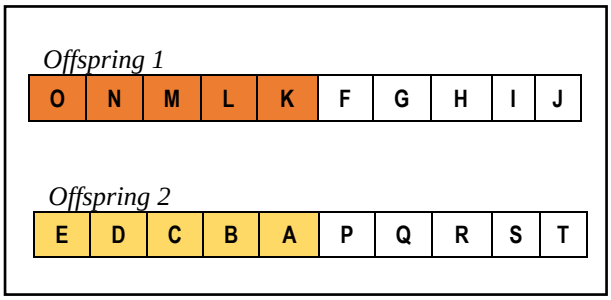


**Fig. 2**. Elements after the crossover process.

## IV. Experimental Evaluation

The performance of the new crossover operator was compared with those of other crossover operators namely, Segmented, Shuffle, Reduced, Surrogate, Half Uniform, Uniform, and Two-Point. For evaluation, processing time was determined using the various operators using a real dataset comprising of 1,000 soil testing and classification results from Cagayan Valley, Philippines. Each object in the dataset represents different soil properties like color, texture, pH level, and mottles encoded into series of binary strings of 0s and 1s.

Testing was done in the same platform using C# programming language to obtain fair comparison. Simulations were performed using a desktop computer with an Intel Core i5 processor with 2.7 GHz processing speed, 4 GB RAM and 500 GB internal memory with 80% free disk space.

The experimental runs were performed using initial population sizes, *N*, of 30, 50, and 100. The data all had lengths, *L*, of 21 bits and the number of generations *G*, was set to 100.

### *G. Comparative Performance*

The performance of the crossover operators were tested for various population sizes in order to determine the effect of population size on processing speed and the results for populations of 30, 50, and 100 are shown in Figures 3, 4, and 5, respectively.

From the figures, it can be seen that PPX consistently has the least processing time among the operators tested regardless of population.

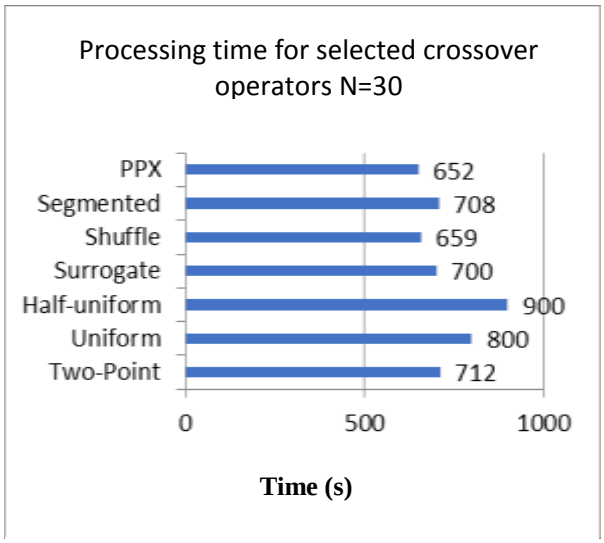


Fig.3. Performance of crossover operators for a population of 30.

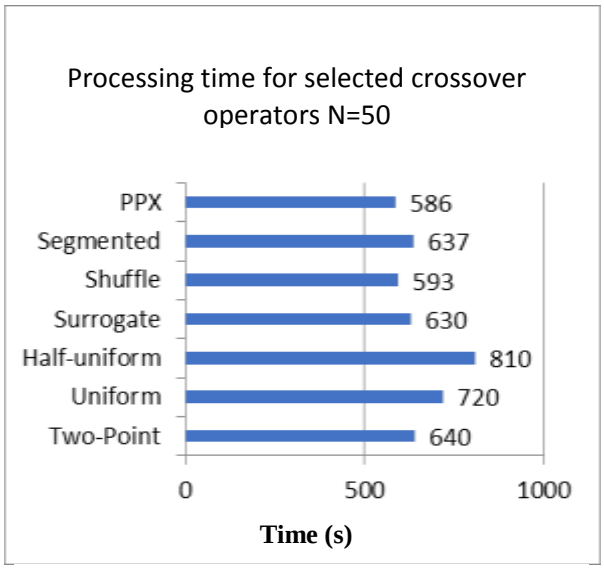


Fig. 4. Performance of crossover operators for a population Of 50.

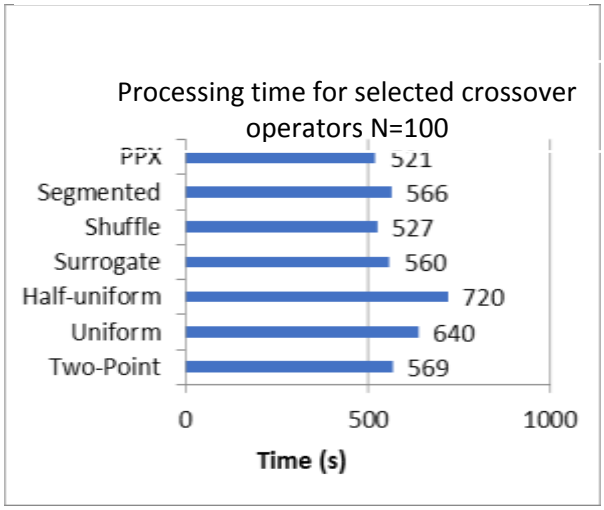


**Fig. 5.** Performance of crossover operators for a population of 100.

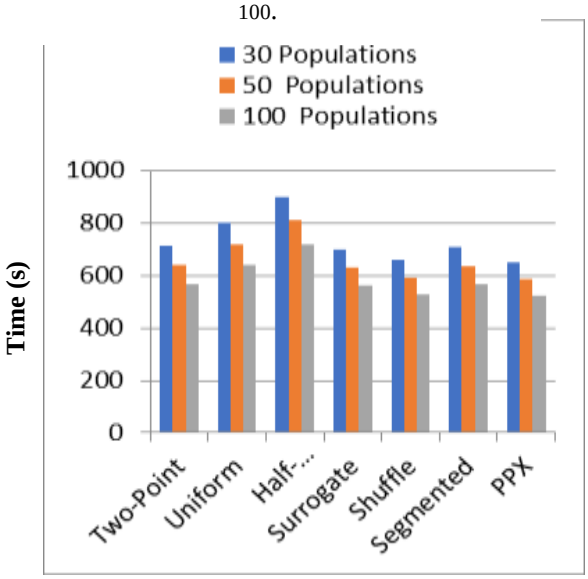


Fig. 6. Effect Of Population Size For Crossover Operator

The obtained numerical results on the effects of number in population are graphically shown in Figure 6. As seen from Figure 6, increasing the population size resulted in an acceleration of computational time.

### *I. Comparative Improvement*

The use of PPX improved the processing time of GA. The improvement over the GA using the existing crossover operators are shown in Table 1. From the table, it can be seen that the performance of PPX is comparable to that of the Segmented operator, showing only 5.6% improvement at population size of 30. Meanwhile, the greatest improvement was observed over the Half-Uniform crossover operator where the improvement reached 24.8% at a population size of 30.

**TABLE 1.**
COMPARATIVE IMPROVEMENT OF GA USING PPX IN TERMS OF PROCESSING TIME OVER EXISTING ALGORITHMS.

| Crossover Operator | Improvement (%) | | |
|---|---|---|---|
| | $N = 30$ | $N = 50$ | $N = 100$ |
| Two-Point | 6 | 5.4 | 4.8 |
| Uniform | 14.8 | 13.4 | 11.9 |
| Half-Uniform | 24.8 | 22.4 | 19.9 |
| Reduced Surrogate | 4.8 | 4.4 | 3.9 |
| Shuffle | 0.7 | 0.7 | 0.6 |
| Segmented | 5.6 | 5.1 | 4.5 |

## V. CONCLUSION

A conclusion section is usually required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions.

## REFERENCES


[1] J. H. Holland: Adaptation in Natural and ArtificialSystems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Cambridge, USA: The MIT Press, 1992

[2] M. Mitchell: An Introduction to Genetic Algorithms. Cambridge, USA: The MIT Press, 1999

[3] D. Dumitrescu, B. Lazzerini, L. C. Jain and A. Dumitrescu: Evolutionary Computation. Florida, USA: CRC Press, 2000

[4] M. Golub: Genetski algoritam: Prvi dio. University of Zagreb, Croatia: Faculty of Electrical Engineering and Computing, 2004

[5] R. L. Haupt and S. E. Haupt: Practical genetic algorithms, second edition. New Jersey, USA: Wiley-Interscience, A John Wiley & Sons, 2004

[6] J. H. Holland: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. Cambridge, USA: The MIT Press, 1992

[7] Z. Michalewitz: Genetic Algorithms + Data Structures = Evolution Programs, third edition. Berlin Heidelberg New York, USA: Springer-Verlag, 1996

[8] M. Mitchell: An Introduction to Genetic Algorithms. Cambridge, USA: The MIT Press, 1999

[9] S. Picek, M. Golub: The New Negative Slope Coefficient Measure, Proceedings of the 10th WSEAS International Conference on Evolutionary Computing, EC'09, 2009, Prag, Czech Republic, pp. 96–101

[10] S. Picek, M. Golub: Dealings with Problem Hardness in Genetic Algorithms, WSEAS Transactions on Computers, Issue 5, Volume 8, pp. 747–756

[11] S. Picek, M. Golub: On the Efficiency of Crossover Operators in Genetic Algorithms with Binary Representation, Proceedings of the 11th WSEAS International Conference on Neural

[12] Networks (NN '10), the 11th WSEAS International Conference on Evolutionary Computing (EC '10) and the 11th WSEAS International Conference on Fuzzy Systems (FS '10), Iasi, Romania, 2010, pp. 167–172

[13] S. Rana: The Distributional Biases of Crossover Operators, Proceedings of the Genetic and Evolutionary Computation Conference, Morgan Kaufmann Publishers, 1999, pp. 549–556

[14] L. D. Whitley: An Executable Model of a Simple Genetic Algorithm. Foundations of Genetic Algorithms 2, 1992

[15] L. Zhonggang, Z. Liang: A Quantum-Inspired Hybrid Evolutionary Method, Proceedings of the 8th WSEAS International Conference on Applied Computer and Applied Computational Science, Hangzhou, China, 2009, pp.1422–425

[16] Olympia Roeva et al: Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modeling.Proceedings of the 2013 Federated Conference on Computer Science and Information Systems pp. 371–376


**Joe G. Lagarteja** is currently taking up his Doctor in Information Technology and now on dissertation writing at Technological Institute of the Philippines, Quezon City, Philippines and finished her MIT degree at University of La Salette, Santiago City, Philippines in the year 2009. He obtained the Bachelor of Science in Information Technology at the Isabela State University, Echague Isabela, Philippines in the year 2007. He started his teaching profession at Institute of Information and Communication Technology at Isabela State University, Echague, Isabela, Philippines. Currently, He has been designated as the Department and BSIS Program Chairman, and has been extensively involved in Research and Extension of the Institute. He has been tapped by various agencies to perform job relevant to her field of specialization. Her field of interest includes information system and data mining.

**Dr. Bobby D. Gerardo** is currently the Vice President for Administration and Finance and holds a rank as professor VI of West Visayas State University, Iloilo City, Philippines. His dissertation was about Discovering Driving Patterns using Rule-based intelligent Data Mining Agent (RiDAMA) in Distributed Insurance Telematic Systems. He has published more than 65 research papers in national and international journals and conferences. He is a referee to international conferences and journal publications such as IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Knowledge and Data Engineering and Elsevier Journal on Pervasive and Mobile Computing, and Ecological Informatics. His research interests lie in the area of distributed systems, telematics systems, CORBA, data mining, we services, ubiquitous computing and mobile communications.

**Dr. Ruji P. Medina** is Dean of the Graduate Programs and concurrent Chair of the Environmental and Sanitary Engineering Program of the Technological Institute of the Philippines in Quezon City. He holds a Ph.D. in Environmental Engineering from the University of the Philippines with sandwich program at the University of Houston, Texas. He counts among ntal modeling and mathematical modeling using multivariate analysis.