United International Journal for Research & Technology
Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

Further Enhancement of Genetic Algorithm Using Multiple Crossover and Mutation Operators

Jose Antonio M. Aguila[1], Loise Gabriel C. Gitalado[2], Vivien A Agustin[3], Richard C. Regala[4] and Mark Christopher R. Blanco[5]

[1,2,3,4,5]Computer Science Department, Pamantasan ng Lungsod ng Maynila, Manila, Philippines

Abstract— This research focuses on further enhancing the performance of genetic algorithms through the utilization of multiple crossover and mutation operators. The primary objectives of this study are to improve the effectiveness of genetic algorithms by employing diverse crossover and mutation operators, develop a novel crossover operator to overcome limitations of existing operators, and propose innovative crossover and mutation operators to address local optima issue. The methodology employed in this research centers around the Nurse Scheduling Problem, which involves creating an optimal schedule for a group of nurses considering various constraints. Genetic algorithms are used to solve this problem, with binary encoding representing the nurse schedule. To enhance the algorithm, three crossover algorithms (Poor and Rich Optimization, Golden Search Algorithm, and Prairie Dog Optimization Algorithm) and three mutation algorithms (Chaotic Vortex Search Algorithm, Capuchin Search Algorithm, and Chaos Cloud Quantum Bat Hybrid Optimization) are proposed. The research findings indicate that the combination of Prairie Dog and Capuchin Search algorithms outperforms other combinations, resulting in a significantly lower number of shift violations (only 7 violations). Overall, this research contributes to the field of genetic algorithms by presenting novel crossover and mutation operators that enhance their performance, particularly in tackling the Nurse Scheduling Problem.

Keywords— Genetic Algorithm, Crossover, Mutation, Capuchin Search algorithm, Chaos Cloud Quantum Bat Hybrid Optimization, Chaotic Vortex Search algorithm, Golden Search Optimization algorithm, Poor and Rich Optimization algorithm, Prairie Dog Optimization algorithm.

INTRODUCTION

Smaller healthcare facilities are under pressure to treat patients promptly because of the growing and aging population. To provide proper care, healthcare personnel such as nurses are scheduled based on the demands of patients and work-hour laws. It takes hours of human labor to finish this process [1]. By using a scheduling system to rationally plan and schedule various courses or services, it is possible to quickly fulfill various limitations and, as a result, produce workable solutions [2].

The Genetic Algorithm (GA), a search heuristic algorithm based on the theories of evolution and natural selection, was first developed by John Holland in the 1970s [3]. It has been widely employed in many practical applications and is a significant and efficient way for solving optimization problems [4].

Genetic Algorithm investigates the population of chromosomes, where each chromosome indicates a different candidate solution to the problem that corresponds to it. It includes several operators, including selection, crossover, and mutation. The fitness function is used to make the selection. To produce a better generation of population, these operators are applied to the potential solutions [5]. There are many genetic algorithms that work well when used to solve difficult optimization problems. However, it has been observed that the recent algorithms still have either one or more following problems: a slow convergence rate, low accuracy on specific problems, or slow performance when trying to solve many real-world problems with complex landscapes [6][7][8].

Aiming at the problems of low solution quality, and easily falling into a local optimum, a genetic algorithm that uses multiple crossover operators and multiple mutation operators is proposed. As for the crossover operators, it will be composed of Poor and Rich Optimization algorithm, Golden Search algorithm, and Prairie Dog Optimization Algorithm. The mutation operators that will be used are Chaotic Vortex Search algorithm, Capuchin Search algorithm, & Chaos Cloud Quantum Bat Hybrid Optimization algorithm.

Statement of the Problem

The study aims to solve the following problems: (1) Genetic Algorithm easily being stuck in local optima. (2) Genetic Algorithm having difficulties in handling high-complexity problems, such as those that involve

UIJRT ISSN: 2582-6832
United International Journal for Research & Technology
Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

large datasets or multiple objectives. (3) New crossover and mutation operators for the Genetic Algorithm.

Objectives

The study aims to solve the stated problems with the following objectives: (1) To improve the performance of the Genetic Algorithm by using multiple crossover and mutation operators. (2) To develop a new crossover operator for genetic algorithms that addresses the limitations of existing operators, such as computational complexity, lack of variation in new offspring, and inefficiency for substantial number of nodes, to improve the ability of genetic algorithms to handle high-complexity problems with large datasets and multiple objectives. (3) To propose novel crossover (Poor and Rich Optimization, Golden Search Algorithm, Prairie Dog Optimization Algorithm) and mutation operators (Chaotic Vortex Search Algorithm, Capuchin Search Algorithm, and Chaos Cloud Quantum Bat Hybrid Optimization) that can enhance the performance of genetic algorithms and overcome the limitations of getting stuck in local optima.

Scope and Limitations

The scope of this study is focused on investigating the use of multiple crossover and mutation operators to enhance the performance of genetic algorithms. Poor and Rich Optimization, Golden Search Algorithm, and Prairie Dog Optimization are to be used as crossover operators. Chaotic Vortex Search, Capuchin Search Algorithm, and Chaos Cloud Quantum Bat Hybrid Optimization are to be used as mutation operators.

The study will be applied in healthcare, specifically nurse scheduling problems where the genetic algorithm would be used to optimize nurses' scheduling in a hospital.

The research will consider the constraints of nurse scheduling such as the availability, skills, and qualifications of nurses, and the hospital needs.

The nurse scheduling problem will be used as the benchmark of the proposed genetic algorithm. The study will be conducted using computer simulations and mathematical analysis, and the results will be presented in the form of tables and figures.

The study will be limited to the use of single objective optimization problems and will not consider multi-objective optimization problems.

THEORETICAL FRAMEWORK

Flowchart of Genetic Algorithm

The Genetic Algorithm starts with a large initial population of random solutions; these solutions (individuals) are then encoded in accordance with the present problem, and the fitness function is used to assess everyone's quality. The GA is primarily dependent on three operators: selection operator, crossover operator, and mutation operator [9].

Selection – the process of selecting the fittest parents from the current population to create the next generation. The fittest will be defined based on the problem.

Crossover – the process of creating fresh solutions (offspring) by combining the genetic material of two parent solutions. Crossover works by taking the genetic material from one parent and combining it with the genetic material from the other parent to create a new offspring solution. It is likely that the offspring will inherit the good genes of the parents and will perform better than their ancestors.

Mutation – the process of making small, random modifications to a solution's genetic make-up, which will be superior to their parents.

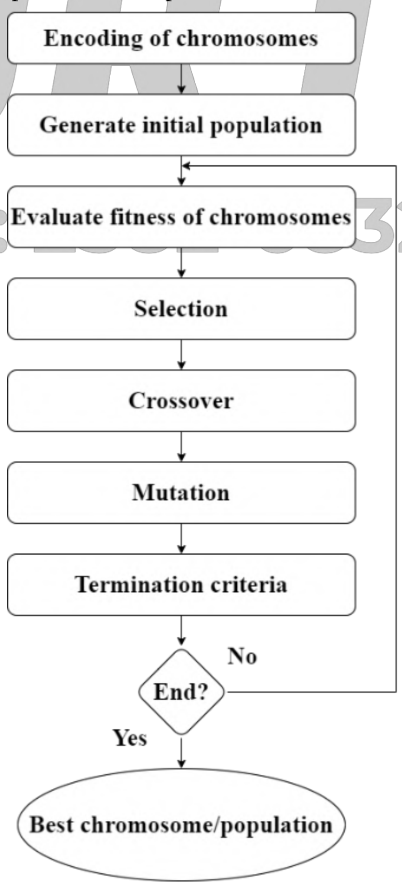

Figure 1.0. *Flowchart of selecting the best mutation algorithm.*

Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

Flowchart of Proposed Genetic Algorithm

The first step in the genetic algorithm process is where the initial population of candidate solutions, represented as chromosomes, is randomly generated. The size of the population and the representation of the chromosomes will depend on the specific problem being solved. The next step is the fitness function. The fitness function is a measure of how well each chromosome represents a solution to the problem. The better the solution, the higher the fitness value. The next step would be selection. This is where the chromosomes with higher fitness values are selected to be used for reproduction. Selection can be done using various selection operators such as roulette wheel selection, tournament selection, etc.

The flowchart of the proposed algorithm adds additional processes for crossover and mutation operators. The crossover operator generates offspring from the selected parents using each crossover algorithm proposed which are Poor and Rich Optimization Algorithm, Golden Search Optimization Algorithm, and Prairie Dog Optimization Algorithm. The offspring from each algorithm will be compared to each other and the best offspring produced will be added to the population.

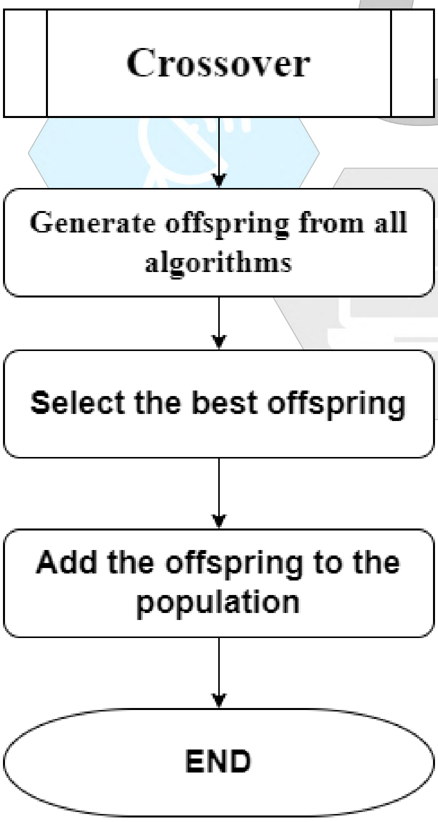

Figure 2.0. *Flowchart of selecting the best crossover algorithm (BC)*

The other algorithm that is also being proposed for the crossover operator will be selecting any crossover algorithm to produce an offspring.

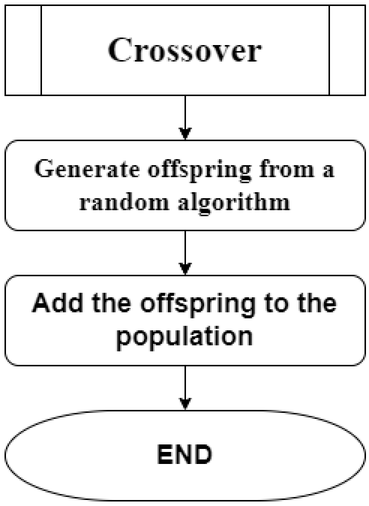

Figure 2.1. *Flowchart of selecting any crossover algorithm (RC).*

After producing offspring from either of the two crossover algorithms, the process will proceed to the mutation operator. The mutation operator generates new individuals from a given population. All mutation algorithms, which are Chaotic Vortex Search Algorithm, Capuchin Search Algorithm, and Chaos Cloud Quantum Bat Hybrid Optimization Algorithm, will generate their own individual, but the best individual will only be added to the population.

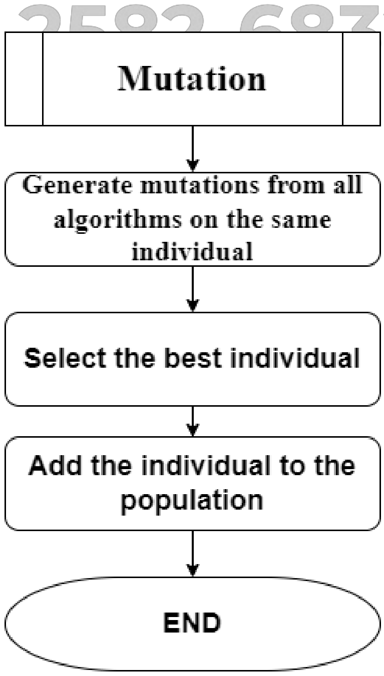

Figure 3.0. *Flowchart of selecting the best mutation algorithm.*

United International Journal for Research & Technology
Volume 04, Issue 07, 2023 | Open Access | ISSN: 2582-6832

The other algorithm that is being proposed also for the mutation operator will be selecting any mutation algorithm to apply mutation to an individual.

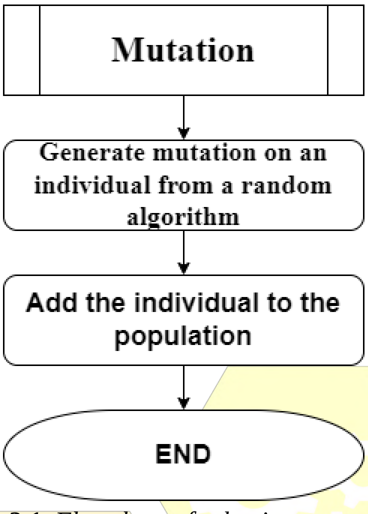

Figure 3.1. *Flowchart of selecting any mutation algorithm.*

After applying mutation to an individual from either of the two mutation algorithms, the process will proceed to the termination criteria. This method is proposed to prevent being stuck from local optimum.

MATERIALS AND METHODOLOGY

This section presents the materials and methodology employed in the research study aimed at enhancing the Genetic Algorithm using multiple mutation and crossover operators using the Python Programming Language. The algorithm under investigation serves as the cornerstone of the research, and its improvement is crucial for addressing specific challenges and achieving desired outcomes. The Nurse Scheduling Problem will be used as a benchmark for the baseline and the proposed Genetic Algorithm.

Materials

Python Programming Language – Python is a versatile, readable, and well-supported language. It provides a user-friendly and efficient way to interact with computers and build software solutions.

DEAP Package – DEAP is a powerful Python package used for implementing and executing evolutionary algorithms. It provides a flexible framework for developing and experimenting with various evolutionary algorithms and computation techniques. DEAP offers functionalities for genetic algorithms. It includes tools for creating populations of individuals, defining fitness functions, applying genetic operators (mutation & crossover), and evolving solutions over multiple generations.

NumPy Package – NumPy is a foundational Python library for scientific computing. It provides support for efficient numerical computations, especially with huge arrays and matrices.

Matplotlib.Pyplot Package – A part of matplotlib library, which is widely used for creating visualizations and plots in Python.

Seaborn Package – Seaborn is a Python data visualization library that is built on top of matplotlib. It offers a sophisticated user interface for producing visually appealing and educational statistical visuals.

Methodology

Baseline Genetic Algorithm – The baseline Genetic Algorithm will be used to compare all proposed genetic algorithms that use different crossover & mutation operators. The crossover operator used is the Uniform Partially Matched operator, while the mutation operator is the Flip Bit operator. These operators belong to the DEAP package to create the flow of the genetic algorithm.

Crossover Process – The crossover process plays a vital role as it combines genetic material from parent individuals to generate new offspring with potentially improved characteristics. In this study, three different crossover algorithms were employed to explore diverse search spaces and enhance the exploration and exploitation capabilities of the algorithm. The three algorithms are as follows:

(1) Poor and Rich Optimization Algorithm - The Poor and Rich Optimization Algorithm (PRO) aims to strike a balance between exploration and exploitation. It employs a dynamic probability distribution mechanism to adjust the search behavior during crossover, with a focus on maintaining diversity while converging towards promising solutions. PRO was implemented to introduce novel genetic information and improve the diversity of the offspring population [10].

(2) Golden Search Optimization Algorithm - The Golden Search Algorithm (GSO) draws inspiration from the mathematical concept of the golden ratio. It utilizes a search mechanism that mimics the golden ratio to guide the crossover process. By dividing the search space based on the golden ratio, GSO efficiently explores the search space while gradually converging towards optimal solutions. GSO was employed to enhance the exploitation capabilities of the algorithm and guide the generation of high-quality offspring [11].

(3) Prairie Dog Optimization Algorithm - The Prairie Dog Optimization Algorithm (PDO) is a nature-inspired algorithm that mimics the behaviors of prairie dogs in a colony. PDO leverages the concept of spatial awareness and cooperation among individuals to guide the crossover process. It utilizes a mechanism that encourages sharing and exchanging genetic information among individuals within the population. PDO was incorporated to promote information exchange and diversity in the offspring generation [12].

Mutation Process – The mutation process introduces random variations into individuals to maintain diversity and potentially discover novel solutions. In this research, three different mutation algorithms were utilized to enhance the exploration capabilities and inject diversity into the population. The following mutation algorithms were used:

(1) Chaotic Vortex Search Algorithm - The Chaotic Vortex Search Algorithm (CVSA) harnesses the concept of chaotic dynamics to introduce random perturbations in the search process. CVSA utilizes chaotic maps to determine the mutation step size and direction, enabling a wide exploration of the search space. By incorporating CVSA, the algorithm introduces chaotic behavior to enhance exploration and discover new regions of the search space [13].

(2) Capuchin Search Algorithm - The Capuchin Search algorithm (CS) is inspired by the intelligent foraging behavior of capuchin monkeys. CS incorporates a dynamic search strategy that balances local exploitation and global exploration. It utilizes a combination of local search operators and random perturbations to guide the mutation process. CS was employed to promote a diverse search and inject novel genetic information into the population [14].

(3) Chaos Cloud Quantum Bat Hybrid Optimization - The Chaos Cloud Quantum Bat Hybrid Optimization (CCQBHO) algorithm combines the chaotic behavior of chaotic maps with the echolocation-inspired search mechanism of bat algorithms. CCQBHO incorporates chaotic maps to dynamically adjust the exploration and exploitation trade-off during mutation. By integrating CCQBHO, the algorithm enhances both exploration and exploitation capabilities, facilitating the discovery of optimal or near-optimal solutions [15].

The Nurse Rostering Problem is defined as the task of creating an optimal schedule for a group of nurses, considering various constraints such as shift requirements, nurse preferences, and labor regulations. The performance of the GA is evaluated by measuring the effectiveness parameters (Consecutive Shift Violations, Shifts Per Week Violations, Nurses Per Shift Violations, and Shift Preference Violations) for the final schedules generated. The results of the parameters are compared across multiple runs of the GA approach. The effectiveness of the chosen crossover and mutation algorithms will be measured based on the minimized violations and if it produced a high-quality schedule.

RESULTS AND DISCUSSION

This section presents the findings of the study and provides a comprehensive analysis and interpretation of the collected data. Through careful examination of data, we aim to answer questions on the various aspects of the research problem and draw conclusions that are supported by evidence.

Table 1.0 *Violations of every Genetic Algorithm variation*

	BGA	PRO + CVSA	PRO + CS	PRO + CCQBHO	GSO + CVSA	GSO + CS	GSO + CCQBHO	PDO + CCQBHO	PDO + CS	PDO + CVSA
Consecutive shift violations	0	2	1	18	0	0	19	2	0	1
Weekly Shifts	[5, 5, 5, 4, 5, 6, 4, 4]	[1, 4, 5, 4, 5, 5, 7, 6]	[4, 5, 4, 3, 5, 5, 5, 5]	[1, 0, 5, 8, 7, 10, 9, 8]	[5, 4, 5, 5, 3, 5, 6, 5]	[5, 5, 3, 5, 4, 5, 5, 5]	[5, 8, 7, 8, 8, 7, 7, 9]	[5, 5, 2, 5, 5, 5, 3, 4]	[4, 5, 5, 5, 5, 5, 5, 5]	[5, 5, 5, 5, 5, 3, 4, 5]
Shifts Per Week Violations	1	3	0	17	1	0	19	0	0	0

Nurses Per Shift	[2, 2, 1, 1, 4, 1, 2, 3, 1, 2, 3, 2, 2, 1, 1, 1, 3, 1, 2, 2, 1]	[3, 3, 1, 2, 2, 1, 1, 2, 2, 1, 2, 1, 3, 2, 0, 2, 2, 1, 2, 2, 2]	[2, 2, 1, 2, 2, 1, 3, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1]	[3, 3, 1, 1, 3, 1, 3, 2, 3, 3, 2, 3, 2, 3, 2, 3, 1, 3, 2, 2, 2]	[2, 2, 1, 3, 2, 1, 2, 1, 1, 2, 1, 2, 2, 3, 1, 2, 3, 1, 2, 2, 2]	[2, 2, 1, 2, 2, 1, 2, 3, 1, 2, 2, 2, 2, 2, 1, 2, 2, 1, 2, 2, 1]	[2, 3, 4, 1, 3, 1, 3, 2, 3, 2, 5, 4, 3, 3, 3, 2, 3, 4, 3, 1, 4]	[1, 2, 1, 2, 2, 1, 1, 2, 1, 3, 2, 1, 2, 2, 1, 2, 1, 1, 3, 2, 1]	[2, 2, 1, 2, 2, 2, 2, 2, 1, 3, 2, 1, 2, 4, 1, 2, 2, 1, 2, 2, 1]	[2, 2, 1, 2, 2, 1, 3, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 1, 2, 2, 2]
Nurses Per Shift Violations	3	3	0	5	2	0	13	3	0	0
Shift Preference Violations	14	16	14	19	11	10	22	10	7	15

Table 1 shows all violations of every variation of the genetic algorithm, including the baseline genetic algorithm. The genetic algorithm that uses the Prairie Dog Optimization algorithm as the crossover operator and the Capuchin Search algorithm as the mutation operator, has the best performance of all variations. It has only 7 shift preference violations, 0 nurses per shift violations, 0 shift per week violations, and 0 consecutive shift violations. The genetic algorithm that uses the Golden Search algorithm as the crossover operator and the Chaos Cloud Quantum Bat Hybrid algorithm as the mutation operator, has the worst performance of all variations. It has 22 shift preference violations, 13 nurses per shift violations, 19 shift per week violations, and 19 consecutive shift violations. There are 5 out of 9 variations of the genetic algorithms that surpass the performance of the baseline genetic algorithm, and there are 2 out of 9 variations that work nearly as well as the baseline genetic algorithm.

CONCLUSION AND RECOMMENDATION

After evaluating various combinations of crossover and mutation algorithms in the context of the Genetic Algorithm, the combination of Prairie Dog Optimization for crossover and Capuchin Search for mutation outperformed the other combinations. Not only did it outperform the other combinations, but it also showed better results compared to the baseline Genetic Algorithm. One of the key measures of performance was the number of violations observed in the generated schedules. The combination of Prairie Dog Optimization for crossover and Capuchin Search for mutation demonstrated the lowest number of violations among all the combinations, with only 7 violations. This indicates that this combination was more effective in satisfying the constraints and preferences of the scheduling problem. The low number of violations suggests that the Prairie Dog Optimization crossover and Capuchin Search mutation algorithms have successfully improved the overall quality of the schedules generated by the Genetic Algorithm. The combination leveraged the strengths of both algorithms, potentially leading to a better exploration and exploitation of the search space, resulting in more optimal solutions.

There are also 4 more combinations of both crossover and mutation operator that exceeded the performance of the baseline Genetic Algorithm which are the (1) Golden Search Optimization crossover and Capuchin Search mutation algorithms, the (2) Prairie Dog Optimization crossover and Chaos Cloud Quantum Bat Hybrid mutation algorithms, the (3) Golden Search Optimization crossover and Chaotic Vortex Search mutation algorithms, and the (4) Prairie Dog Optimization crossover and Capuchin Search mutation algorithms.

Therefore, it can be concluded from the results that the most effective variation for resolving scheduling issues is the combination of Prairie Dog Optimization as the crossover operator and Capuchin Search as the mutation operator. Additionally, the Golden Search Optimization algorithm can improve the performance of the Genetic Algorithm as the crossover operator. Furthermore, the Chaotic Vortex Search algorithm can also improve the performance of the Genetic Algorithm as the mutation operator. The proponents would like to recommend for the future researchers to test or apply all the promising combinations of crossover and mutation operators in the enhanced genetic algorithm to different problems such as the travelling salesman problem and the knapsack problem to test the performance of the algorithm in different perspectives. It is also recommended to compare the enhanced genetic algorithm to other algorithms for future studies and to apply the enhanced genetic algorithm to real-world scenarios.

The proponents would like to recommend for the future researchers to test or apply all the promising combinations of crossover and mutation operators in the enhanced genetic algorithm to different problems such as the travelling salesman problem and the knapsack problem to test the performance of the algorithm in different perspectives. It is also recommended to compare the enhanced genetic algorithm to other algorithms for future studies and to apply the enhanced genetic algorithm to real-world scenarios.

REFERENCES

[1] Schrack, J., Ortega, R., Dabu, K. S., Truong, D. D., Aibin, M., & Aibin, A. (2021). Combining Tabu Search and Genetic Algorithm to Determine Optimal Nurse Schedules. https://doi.org/10.1109/ccece53047.2021.9569111

[2] Amindoust, A., Asadpour, M., & Shirmohammadi, S. (2021). A Hybrid Genetic Algorithm for Nurse Scheduling Problem considering the Fatigue Factor. Journal of Healthcare Engineering, 2021, 1–11. https://doi.org/10.1155/2021/5563651

[3] Hassanat, A., Alkafaween, E., Alnawaiseh, N., Abbadi, M., Alkasassbeh, M., & Alhasanat, M. (2016). Enhancing Genetic Algorithms using Multi Mutations: Experimental Results on the Travelling Salesman Problem. International Journal of Computer Science and Information Security. 14. 785-801

[4] Sun, N., & Lu, Y. (2019). A self-adaptive genetic algorithm with improved mutation mode based on measurement of population diversity. Neural Computing and Applications. 31. 1-9. 10.1007/s00521-018-3438-9.

[5] Akter, S., Nahar, N., ShahadatHossain, M., & Andersson, K. (2019). A New Crossover Technique to Improve Genetic Algorithm and Its Application to TSP. https://doi.org/10.1109/ecace.2019.8679367.

[6] Mahjoob, M., Fazeli, S. a. S., Milanlouei, S., Tavassoli, L. S., & Mirmozaffari, M. (2021). A modified adaptive genetic algorithm for multi-product multi-period inventory routing problem. Sustainable Operations and Computers, 3, 1–9. https://doi.org/10.1016/j.susoc.2021.08.002

[7] Su, Y., Guo, N., Tian, Y., & Zhang, X. (2019). A Non-Revisiting Genetic Algorithm Based on a Novel Binary Space Partition Tree. Information Sciences. 512. 10.1016/j.ins.2019.10.016.

[8] Zhang, P., Wang, J., Tian, Z., Sun, S., Li, J., & Yang, J. (2022). A genetic algorithm with jumping gene and heuristic operators for traveling salesman problem. Applied Soft Computing, 127, 109339. https://doi.org/10.1016/j.asoc.2022.109339

[9] Hassanat, A. B. A., & Alkafaween, E. (2017). On enhancing genetic algorithms using new crossovers. International Journal of Computer Applications in Technology, 55(3), 202. https://doi.org/10.1504/ijcat.2017.10005868

[10] Moosavi, S. G., & Bardsiri, V. K. (2019). Poor and rich optimization algorithm: A new human-based and multi populations algorithm. Engineering Applications of Artificial Intelligence, 86, 165–181. https://doi.org/10.1016/j.engappai.2019.08.025

[11] Noroozi, M. J., Mohammadi, M. R., Efatinasab, E., Lashgari, A., Eslami, M., & Khan, B. (2022). Golden Search Optimization Algorithm. IEEE Access, 10, 37515–37532. https://doi.org/10.1109/access.2022.3162853

[12] Ezugwu, A. E., Agushaka, J. O., Abualigah, L., Mirjalili, S., & Gandomi, A. H. (2022). Prairie Dog Optimization Algorithm. Neural Computing and Applications, 34(22), 20017–20065. https://doi.org/10.1007/s00521-022-07530-9

[13] Gharehchopogh, F. S., Maleki, I., & Dizaji, Z. A. (2021). Chaotic vortex search algorithm: metaheuristic algorithm for feature selection. Evolutionary Intelligence, 15(3), 1777–1808. https://doi.org/10.1007/s12065-021-00590-1

[14] Braik, M., Sheta, A., & Al-Hiary, H. (2021). A novel meta-heuristic search algorithm for solving optimization problems: capuchin search algorithm. Neural Computing and Applications, 33(7), 2515–2547. https://doi.org/10.1007/s00521-020-05145-6

[15] Li, M., Wang, Y., Geng, J., & Hong, W. (2021). Chaos cloud quantum bat hybrid optimization algorithm. Nonlinear Dynamics, 103(1), 1167–1193. https://doi.org/10.1007/s11071-020-06111-6