International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

Hybrid Simulated Annealing: An Efficient Optimization Technique

Ankita[1], Rakesh Kumar[2]
Research Scholar[1],
Department of Computer Science & Applications,
Kurukshetra University, Kurukshetra,
Kurukshetra, India
ankita.30@kuk.ac.in[1],
Professor[2]
Department of Computer Science & Applications,
Kurukshetra University, Kurukshetra
Kurukshetra, India
rakeshkumar@kuk.ac.in[2]

Abstract: Genetic Algorithm falls under the category of evolutionary algorithm that follows the principles of natural selection and genetics, where the best adapted individuals in a population are more likely to survive and reproduce, passing on their advantageous traits to their offsprings. Crossover is a crucial operator in genetic algorithms as it allows the genetic material of two or more individuals in the population to combine and create new individuals. Optimizing it can potentially lead to better solutions and faster convergence of the genetic algorithm. The proposed crossover operator gradually changes the alpha value as the search proceeds, similar to the temperature in simulated annealing. The performance of the proposed crossover operator is compared with the simple arithmetic crossover operator. The experiments are conducted using Python and results show that the proposed crossover operator outperforms the simple arithmetic crossover operator. This paper also emphasizes the importance of optimizing genetic operators, particularly crossover operators, to improve the overall performance of genetic algorithms.

Keywords: Benchmark Test function; Crossover; Hybridization; Optimization; Simulated Annealing.

I. INTRODUCTION

Optimization is a crucial task in different domains, including engineering, finance, economics, and computer science. It involves selecting the best solution from a set of possible alternatives, given certain constraints and objectives [25]. With the increasing complexity of real-world problems, traditional optimization methods may unable to achieve the optimal solution within a reasonable time frame. Therefore, researchers are always looking for new and efficient methods to address complex optimization problems. Metaheuristic algorithms, such as Genetic Algorithm (GA), have emerged as popular approaches to tackle such problems. These algorithms are motivated by natural process, such as evolution, swarm intelligence, and simulated annealing, and are designed to explore a large search space to acquire the most optimal solution. GA is one of the most popular metaheuristic approaches which have been utilized in broad range of optimization problems in diverse fields such as finance, engineering, transportation, biology, and many others. It offers a powerful framework to solve hard optimization problems by imitating the natural evolutionary process of genetics. GA is an effective tool to find solution of complex optimization problems that traditional methods struggle with. By mimicking the process of natural selection, it can efficiently search through an enormous search space to find the optimal solution [6].

Genetic operators play a crucial role in genetic algorithms, which are integral tools for optimization, and are essential for achieving optimal solutions. The selection of the operator depends on the unique problem being addressed and the attributes of the exploration space. The genetic operators include selection, crossover and mutation. Crossover plays a crucial role in genetic algorithms as it generates new solutions by mating existing chromosomes [29]. The quality of GA's solutions is heavily dependent on the performance of the crossover operator, emphasizing the need for its improvement. Efficient crossover operators promote population diversity and facilitate exploration of various solution space regions, resulting in high-quality solutions. Its goal is to produce offspring that have better fitness or higher objective function values than the parent solutions, thus enhance the entire quality of the population over time. The performance of a genetic

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

algorithm can be greatly influenced by the choice of crossover operator to solve a specific problem. Therefore, selecting an appropriate crossover operator is an important consideration in the design of a genetic algorithm. This article intends to investigate renowned crossover techniques and suggest a novel approach that integrates the idea of simulated annealing (SA).

The following is the organization of the paper: Section 2 depicts the views of different researchers to provide a background for the proposed approach. Section 3 describes the different types of crossover used in genetic algorithms. Section 4 of the paper outlines the proposed approach that combines GA and SA. Additionally, section 5 discusses various standard test functions employed to determine the efficiency of the hybrid methodology. Section 6 contains the experimental findings and analysis, which will likely be an important part of the paper. At last, the paper is briefly concluded in section 7.

II. RELATED WORK

Genetic algorithm emulates genetic principles and natural selection to determine the most efficient solution for a problem using search technique. It can discover the best solution to a problem by imitating the process of natural evolution. The paper describes the entire workflow of GA, including population initialization, fitness evaluation, parent's selection, crossover of parents to produce offsprings, mutation to induce variety in population, and replacement of previous population with new updated population. In addition, the paper describes different types of operators used in GA, their applicability in various domains, and the advantages of using GA as a solution approach. The challenges faced by GA to solve optimization problems are also addressed, which is an important aspect to consider when using GA as a solution approach. Overall, the paper provides a comprehensive overview of GA and its features. [1][8][15][26][18]

The crossover operator is an important component of GA that is used to explore new solutions by exchanging sub-strings between selected parents. The papers surveys prevailing crossover approaches and propose new approaches or merge together multiple types to enhance the effectiveness of GA. Moreover, the effectiveness of different crossover methods is analyzed and compared using benchmark problems or real-world applications to identify which technique works best in which scenario. Some proposed crossover techniques, such as ring crossover and heuristic crossover are discussed and their performance is evaluated. Exploring and improving crossover techniques are an important aspect of optimizing the performance of genetic algorithms. [4][21][29][20][22]

SA is a search strategy that is used to find global optima by escaping local optima. The papers provide a concise overview of SA, including its history and motivation for its popularity. It is also discussed how SA can solve broad range of optimization problems in various domains, including engineering, finance, transportation, and logistics, among others. SA is a useful tool for optimizing complex problems and has a wide range of applications across various fields. Its ability to escape local optima makes it a valuable addition to the suite of optimization techniques available to researchers and practitioners.[5][12][17][9]

Memetic Algorithm (MA) is a sort of hybrid evolutionary algorithm that merges the principles of traditional genetic algorithms with local search approaches, such as hill climbing or tabu search. MAs are actually designed to find good quality, precise solutions to complex optimization problems by combining global search with local search methods. It has been discussed how memetic algorithms represent a futuristic research direction to develop new hybridization techniques and apply MAs to emerging problems in fields such as transportation, finance, and bio-informatics. The papers explore the specific applications and advantages of memetic algorithms in more detail, and discuss their potential for advancing optimization research in the future. [3][28][14]

The hybridization of SA with other metaheuristic techniques is explored to gain benefit of the strengths of various algorithms. Specifically, the paper mentions the hybridization of SA with Cuckoo Search, Genetic Algorithm, ACO (Ant Colony Optimization) to sort out optimization problems. Additionally, the proposed approach is evaluated on benchmark instances through experimentation to assess its performance. Subsequently, the potency of the hybrid approach is evaluated in contrast to other metaheuristic methods, including but not limited to genetic algorithms, particle swarm optimization, and ant colony optimization. The experimental results are demonstrated in the form of graphs, tables, or statistical tests, which allow for a quantitative comparison of the performance of different approaches. The results proved that the hybrid approach outperforms the simple metaheuristic approaches with reference to convergence speed, quality of solution and robustness. Overall, the paper highlights the potential benefits of combining different optimization techniques through hybridization, and provides evidence for the effectiveness of this approach in solving complex optimization problems. [24][30][13]

III. CROSSOVER

Crossover is a genetic operator which is used to generate new solutions by merging two or more existing solutions [21]. It is commonly used in genetic algorithms, but can also be used in other optimization techniques. In GA, crossover is an

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

operator which is used for generating new candidate solutions. It is based on the concept of genetic recombination and is influenced by the natural process of genetic crossover. The resulting offspring can exhibit new combinations of genetic information that may not have been present in the original population, which leads to rise in diversity. However, the effectiveness of crossover depends on the nature of the problem being solved, as well as the chosen crossover operator and parameters. The basic idea of crossover in genetic algorithms is to consider two parents and generate new off-springs by combining parts of each parent. The genetic material of two parents is combined to create genetically diverse offspring. This occurs through a process called recombination, which involves the exchange of genetic material between homologous chromosomes.

Following are some crossover operators in genetic algorithm:

- **1-point crossover:** One-point crossover is a operator used in GA to create new off-springs from two parents. It is a type of crossover operator that involves selecting a single point along the chromosome of the parent solutions and exchanging the hereditary material between the two parents to create two new off-springs. It involves choosing a random position on the chromosome of the parent solutions as the crossover point. The genetic material situated after the crossover point of one parent is replaced with the genetic material situated after the crossover point of the other parent to generate new offsprings.

- **2-point crossover:** Two-point crossover is used in GA to produce new off-springs from two parents. It involves selecting two points along the chromosome of the parent solutions, due to which, new offsprings are generated by crossing over between those parents. It selects two crossover points randomly along the chromosome of the parent solutions. Then, the hereditary material within the two crossover points of the first parent is replaced with the hereditary material within the two crossover points of the other parent to generate new descendants.

- **Uniform crossover:** Uniform crossover is used in GA to produce new off-springs from two parents. It randomly selects genetic information from the parents to create new off-springs. Its process involves randomly selecting a binary mask of 0s and 1s, where a 0 indicates that the corresponding bit will be taken from the first parent solution, and a 1 indicates that the corresponding bit will be taken from the second parent solution. The binary mask is applied to each bit position along the chromosome of the parents to produce new offsprings.

- **Arithmetic crossover:** Arithmetic crossover is a type of crossover operator that involves computing the arithmetic mean of the corresponding gene values in the parent solutions to create new gene values for the offspring solutions. The process of arithmetic crossover involves selecting two parent solutions with numerical values as genes. For each gene position along the chromosome, the arithmetic mean of the corresponding gene values in the parent solutions is computed to create a new gene value for the offspring solutions.

- **Partially Mapped crossover:** Partially Mapped Crossover (PMX) involves swapping a fragment of genes between the parents to produce new off-springs. The process of PMX involves selecting two parent solutions with permutations as genes. For each gene position along the chromosome, a segment of genes is randomly selected from the parent solutions. The selected segment is then swapped between the parents to generate new off-springs. The genes outside of the selected segment are then mapped to the corresponding positions in the other parent solution.

- **Order crossover:** Order Crossover (OX) involves swapping a segment of genes between the parents to generate new off-springs, while maintaining the relative order of the genes. The process of OX involves selecting two parent solutions with permutations as genes. For each gene position along the chromosome, a segment of genes is randomly selected from one of the parent solutions. The selected segment is then copied to the offspring solution, maintaining the relative order of the genes. The remaining genes are then filled in from the other parent solution chronologically, without duplicating any of the genes already copied to the offspring solution.

- **Cycle crossover:** Cycle Crossover (CX) is a genetic operator that involves creating a cycle of genes between the parents to create off-springs for next generation. The process of CX involves selecting two parent solutions with permutations as genes. One of the parents is selected at random to provide the first gene from the offspring. The gene at the same position in the other parent solution is then located, and a cycle is created by tracing the corresponding genes in both parent solutions. The genes in the cycle are then copied to the offspring solution from the first parent, and the remaining genes are then sequentially inherited from the other parent along the route.

IV. PROPOSED APPROACH

a) Hybrid Genetic Algorithm

Hybridization is an approach used in genetic algorithms to combine varied genetic operators to boost algorithmic performance [11]. It is a powerful technique in genetic algorithms and has been shown to upgrade the proficiency and potential of the algorithm in broad applicability. However, the choice of which operators and techniques to combine ma have

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

a considerable significance on the algorithm's efficiency, and must be selected and tested carefully.

A hybrid genetic algorithm is an optimization algorithm that combines genetic algorithms with other optimization techniques, such as local search or gradient descent, as a strategy for optimization. In particular, a hybrid genetic algorithm may use mathematical expressions or other techniques to improve the effectiveness of the crossover process, which is a key component of genetic algorithms. By combining multiple optimization techniques, a hybrid genetic algorithm can better navigate the search space and find optimal or near-optimal solutions more efficiently. The primary reason for hybridization of metaheuristics is to combine the strengths of different metaheuristic algorithms and take advantage of their common aspects to achieve better optimization results [10]. In a hybrid genetic algorithm, additional optimization techniques are combined with the genetic algorithm to optimize its performance. For example, local search techniques can be used to refine the solutions produced by the genetic algorithm helps to avoid getting stuck in local optima. Similarly, gradient descent can be used to increase the pace of convergence of the algorithm.

The goal of a hybrid GA is to take advantage of the strengths of both the genetic algorithm and the additional optimization technique, while minimizing their weaknesses. This can lead to improved solution quality and faster convergence rates compared to using either technique in isolation.

b) *Simulated Annealing*

It is a search method that is utilized to discover the global minimum of a function, particularly in the area of combinatorial optimization problems. The inspiration for this technique is derived from the annealing process used in metallurgy, in which a substance is heated and gradually cooled to attain a state of minimum energy [5]. In the process of exploring the solution space, the algorithm can accept worse solutions with a certain probability controlled by a temperature parameter. This permits the algorithm to escape local optima and continue exploring the solution space with a hope to find better global optimum. Initially, the algorithm adopts a solution and then gradually alters it by introducing random disturbances. The new solution is then evaluated to determine whether it is better or worse than the current solution. If it is better, the new solution is accepted as the current solution. However, if it is worse, the algorithm may still accept it with a certain probability based on a temperature parameter.

At the start of the algorithm, the temperature parameter is high, allowing the algorithm to accept worse solutions more easily. As the algorithm progress, the temperature is gradually decreased, making it less likely that the algorithm will accept worse solutions. This gradual cooling process allows the algorithm to escape local optima and explore the solution space more thoroughly.

The key advantage of simulated annealing is that it can escape local optima and find the global optimum, even in cases where the function being optimized is non-convex or has multiple local optima. However, the algorithm can be computationally expensive, particularly for high-dimensional problems, and the choice of the temperature schedule and probability distribution can affect the algorithm's performance. Simulated annealing has been used in a wide range of applications, including optimization problems in physics, chemistry, engineering, finance, and machine learning. The algorithm has also been combined with other optimization techniques, such as genetic algorithms, to create hybrid optimization algorithms that can further improve performance.

c) *Proposed Crossover*

Crossover is a pivotal operator in genetic algorithms as it generates new off-springs from two or more parents. The effectiveness of the crossover operator can depend on the encoding scheme used, and it is possible that a particular crossover may generate optimal solution in one encoding scheme but can produce incompatible solution in another encoding scheme. Here, an interesting proposal is given for a new crossover operator that incorporates both arithmetic crossover and the concept of simulated annealing. By gradually decreasing the value of alpha with each generation, the algorithm becomes more focused on exploiting promising areas of the solution space and is unlikely to get trapped in local optima. The gradual decrease in alpha can be thought of as the cooling process in simulated annealing, where the temperature parameter is reduced over time to allow the system to settle into a low-energy state. Consequently, the value of alpha in arithmetic crossover is decreasing with each iteration to guide the search towards the global minimum. Initially, the genetic algorithm sets the value of alpha to 0.9, which remains constant throughout the starting phase. Subsequently, alpha is gradually reduced by 0.2 after each generation until it reaches 0.1. At this point, alpha is reset back to its initial value of 0.9. The entire procedure is replicated for a designated number of generations. This reset allows the algorithm to periodically re-explore the solution space with a higher degree of exploration, possibly detecting unexplored portions of the search space that were missed during earlier iterations.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

V. TEST FUNCTIONS

Benchmark test functions are mathematical functions that are generally used to access and compare the proficiency of optimization algorithms. These functions are typically designed to be simple, well-defined, and have known global minima, making them useful for testing and comparing different optimization algorithms. These are commonly employed in the optimization community to evaluate algorithm performance with reference to their speed of convergence, accuracy, robustness, and scalability. Different types of benchmark test functions exist, each with their own unique characteristics and properties. Table 1 displays the titles of the different benchmark functions used in this article.

Function	Name
F1	Sphere
F2	Rosenbrock
F3	Rastrigin

Table 1: Benchmark Functions

a) Sphere Function

This is a straightforward and frequently employed evaluation function in the process of optimization. The principal aim of this function is to accomplish the potential of optimization models to explore the global minimum of a function in a space with many dimensions, as a mission. It is convex, continuous, and differentiable, making it a simple and tractable function to optimize. Two-dimensional form of sphere function is shown in Figure 1.

Mathematical expression:

$$F1(x) = \sum [x_i^2]$$

$$-5.12 <= x_i <= 5.12$$

global minimum: fn(x)=0, xi=0

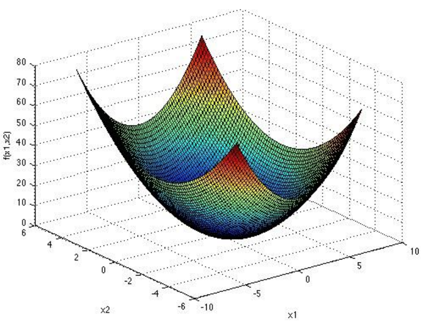

Figure 1: Sphere Function

b) Rosenbrock Function

This function is also referred to as Rosenbrock's valley or the banana function, is a well-known test function used in optimization to access the performance of optimization algorithms. It is designed to achieve the objective of locating the minimum value of an objective function which is continuous and has real values. It has several local minima, which can pose a challenge for optimization algorithms that rely on gradient-based methods. To find the global minimum, algorithms must traverse a narrow, parabolic-shaped valley. Identifying the relevance of the parabolic valley can be a complex task, but this can be resolved through the application of the problem conjunction technique [2].

Mathematical expression:

$$F2(x) = \sum [b(x_{i+1} - x_i^2)^2 + (a - x_i)^2]$$

$$-2.048 <= x_i <= 2.048$$

The numerical values of the constant parameters 'a' and 'b' are typically set at fixed values of 1 and 100, respectively.

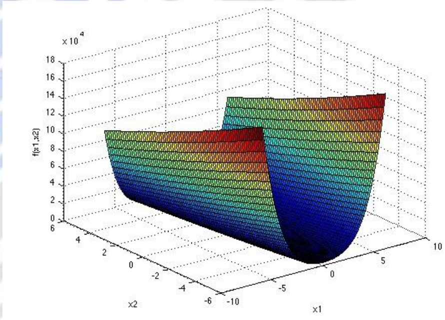

Figure 2: Rosenbrock function

c) Rastrigin Function

It is a multi-modal, non-convex function possessing numerous local valleys. This is commonly employed as a benchmark to examine the effectiveness of optimization models by dealing with noise, randomness, and multiple local minima. It is typically applied to analyze the capability of various optimization algorithms to handle search spaces with high dimensions, where the number of variables can be large. It can be easily scaled to any dimensionality.

Mathematical Expression:

$$F3(x) = 10 * n + \sum [x_i^2 - 10 * cos(2 * 3.14 * x_i)]$$

$$-5.12 <= x_i <= 5.12$$

global minimum: fn(x)=0, xi=0

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

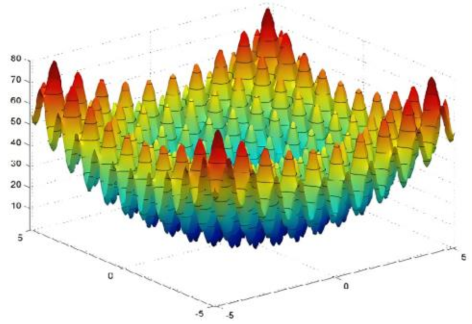

Figure 3: Rastrigin function

VI. FINDINGS AND ANALYSIS

a) Experimental Setup

In this paper, a detailed description of the implemented Python code has been provided for evaluating the proposed arithmetic crossover. The use of three benchmark functions and the roulette wheel selection method is also a good approach to evaluate the performance of the crossover operator. The use of simulated annealing for crossover and mutation is also an interesting idea, as it can prevent being trapped in local optima. Overall, the description of the implementation process seems clear and detailed.

The process of selecting parents using the roulette wheel selection method involves assessing the performance of each member within the population. Then, after selecting parents, the proposed arithmetic crossover is merged with SA to create new offspring. Mutation is then applied to some of the genes in the offspring to introduce additional variation in the population. Finally, a replacement strategy is used to produce next population for the succeeding generation. The entire procedure is replicated for a designated number of generations or till the objective function has been achieved.

Parameter	**Value**
Size of population	10, 20, 50
Iteration steps	100, 200
Encoding Scheme	Real Encoding
Selection Type	Roulette Wheel Selection
Probability of crossover	0.8
Probability of mutation	0.01

Table 2: Parameter Configuration

The implementation of the work involves the utilization of certain values and parameters, which are presented in Table 2. The population size is set at 10, 20, and 50 for 100 and 200 generations. To assess the effectiveness of the proposed approach, a practical encoding scheme is selected, and various benchmark functions are employed. The crossover probability is held constant at 0.8, while the mutation probability is fixed at 0.01.

b) Results and Observations

This section depicts the outcomes of the executed code. The proposed approach's efficacy is evaluated using three benchmark functions. Graphs are used to compare the minimum fitness scores for different test functions for 100 and 200 generations.

N		**10**	**20**	**50**
Gen=100	Simple crossover	1.166	3.686	4.410
	Hybrid crossover	0.792	2.016	3.648
Gen=200	Simple crossover	0.504	1.562	4.452
	Hybrid crossover	0.168	1.416	3.686

Table 3: Minimum Fitness values for F1 function

N		**10**	**20**	**50**
Gen=100	Simple crossover	393.149	1244.917	378.668
	Hybrid crossover	350.424	871.001	317.922
Gen=200	Simple crossover	405.009	421.249	100.318
	Hybrid crossover	310.610	362.112	88.435

Table 4: Minimum Fitness values for F2 function

N		**10**	**20**	**50**
Gen=100	Simple crossover	-4.468	-0.900	-6.831
	Hybrid crossover	-5.930	-5.047	-8.598
Gen=200	Simple crossover	-6.015	-5.511	-7.277
	Hybrid crossover	-8.999	-5.999	-8.998

Table 5: Minimum Fitness values for F3 function

In the Results and Analysis section, Table 3 shows the fitness values obtained using arithmetic crossover and crossover combined with SA for F1 function. Table 4 shows the minimum fitness values of F2 function, while Table 5 demonstrates the minimum fitness values of F3 function.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

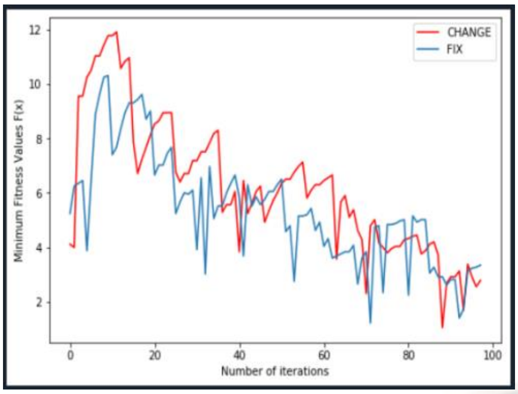

Figure 4: Comparison of Minimum fitness values of F1 for 100 generations (N=10)

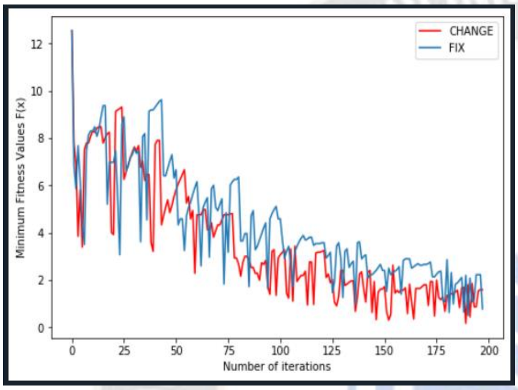

Figure 5: Comparison of Minimum fitness values of F1 for 200 generations (N =10)

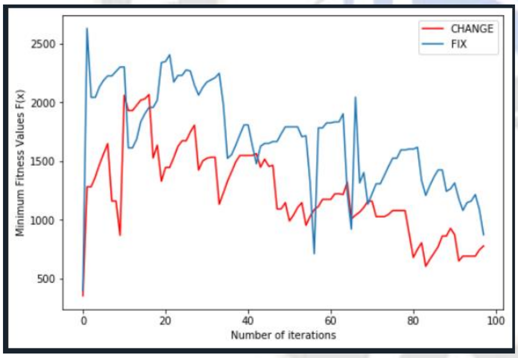

Figure 6: Comparison of Minimum fitness values of F2 for 100 generations (N=10)

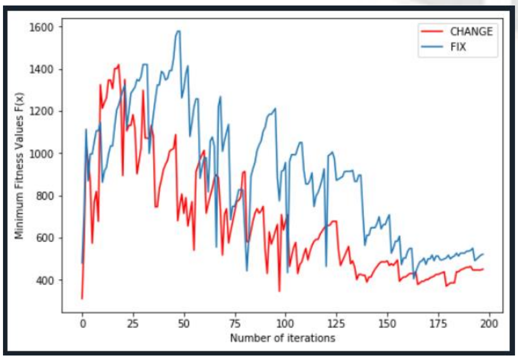

Figure 7: Comparison of Minimum fitness values of F2 for 200 generations (N=10)

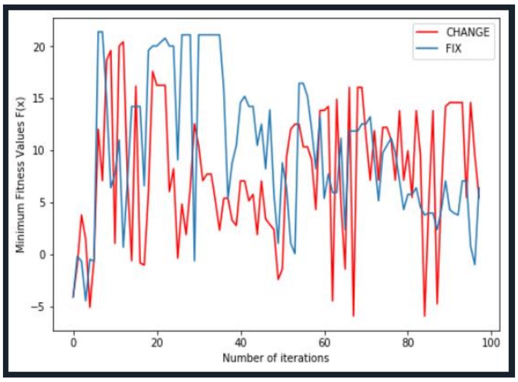

Figure 8: Comparison of Minimum fitness values of F3 for 100 generations (N=10)

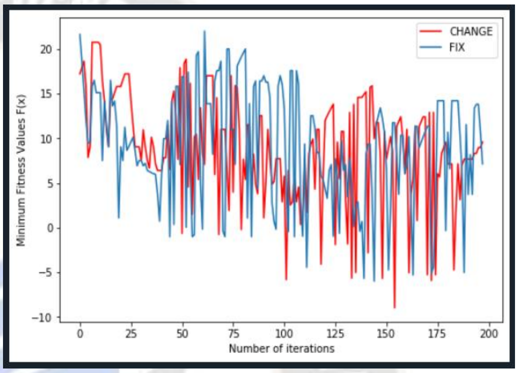

Figure 9: Comparison of Minimum fitness values of F3 for 200 generations (N=10)

Figure 4 and Figure 5 illustrate the effectiveness of GA for the F1 function using the basic crossover operator and modified crossover operator, respectively. A visual comparison of the two operators is made for both 100 and 200 generations.

Within the results visualization segment, Figure 6 and Figure 7 depict the evaluation of the minimum fitness values obtained after 100 and 200 generations of optimization for F2 function. Additionally, Figure 8 and Figure 9 illustrate the fitness values of function F3 for the same number of generations and make a comparison.

Based on the results, it is reported that the modified crossover operator excels the simple crossover operator for the F1 function. This suggests that the proposed modification to the crossover operator is effective in enhancing the performance of the genetic algorithm.

VII. CONCLUSION

Genetic Algorithm is a technique that draws inspiration from genetic evolution's underlying principles. Crossover is a fundamental genetic operation as it generates new solutions by mating existing chromosomes. The proposed crossover operator in this paper introduces a changing value of alpha as

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
DOI: https://doi.org/10.17762/ijritcc.v11i7s.6975
***Article Received**: 31 March 2023 **Revised**: 18 May 2023 **Accepted**: 29 May 2023*

the search proceeds. The alpha value determines the ratio of genetic material that is inherited from each parent during the crossover operation. By gradually changing the alpha value, the proposed crossover operator can explore a wider range of solutions and reduce the risk to stick in local optima. The use of simulated annealing to gradually change the alpha value is a novel and interesting approach to improving the performance of crossover in genetic algorithms. Overall, the proposed crossover operator that incorporates the concept of simulated annealing to adjust the alpha value in each generation has shown promising results in boosting the efficiency of Genetic Algorithm. The analysis performed with three benchmark functions indicated that the proposed crossover method outperforms the traditional crossover technique. This suggests that variations in genetic operators, like crossover, can boost the performance of Genetic Algorithm to solve optimization problems. Further research can explore the potential benefits of incorporating other genetic operators, such as mutation or selection, in combination with the proposed crossover to further improve the proficiency of the genetic algorithm.

References

[1] Manoj Kumar, Mohammad Husain, Naveen Upreti, Deepti Gupta. Genetic algorithm: Review and application. International Journal of Information Technology and Knowledge Management, 2: 451–454, 2010.

[2] Nasib Singh Gill, Kapil Juneja. Optimization of dejong function using GA under different selection algorithms. International Journal of Computer Applications (0975 – 8887), 64: 28–33, 2013.

[3] Pablo Moscato, Carlos Cotta. A modern introduction to memetic algorithms. International Series in Operations Research Management Science, pages 141–183, 2010.

[4] Ahmad B. A. Hassanat, Esra'a Alkafaween. On enhancing genetic algorithms using new crossovers. International Journal of Computer Applications in Technology, 2017.

[5] Alexander G. Nikolaev, Sheldon H. Jacobson. Simulated annealing. 146,2010.

[6] Ankita. Optimization of rosenbrock function using genetic algorithm. Turkish journal of Computer and Mathematics Foundation, 12: 3364– 3368, 2021.

[7] Mr. Rahul Sharma. (2013). Modified Golomb-Rice Algorithm for Color Image Compression. International Journal of New Practices in Management and Engineering, 2(01), 17 - 21. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/13

[8] Ankita. Travelling salesman problem using ga-aco hybrid approach: A review. In ICDSMLA 2020: Proceedings of the 2nd International Conference on Data Science, Machine Learning and Applications, volume 783, pages 105–113, 2021.

[9] Annu Lambora, Kunal Gupta, Kriti Chopra. Genetic algorithm- a literature review. In International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (Com-IT-Con), pages 380– 384, 2019.

[10] Avrim Blum, Chen Dan, Saeed Seddighin. Learning complexity of simulated annealing. In Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, volume 130, 2021.

[11] Christian Blum, Jakob Puchinger, Gunther Raidl and Andrea Roli. Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing, Elsevier, 11: 4135–4151, 2011.

[12] Crina Grosan, Ajith Abraham. Hybrid evolutionary algorithms: Methodologies, architectures, and reviews. Hybrid Evolutionary Algorithms Studies in Computational Intelligence, 75: 1–17, 2007.

[13] Darrall Henderson, Sheldon H. Jacobson, Alan W. Johnson. The theory and practice of simulated annealing. 57: 287–319, 2003.

[14] Ahammad, D. S. K. H. (2022). Microarray Cancer Classification with Stacked Classifier in Machine Learning Integrated Grid L1-Regulated Feature Selection. Machine Learning Applications in Engineering Education and Management, 2(1), 01–10. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/18

[15] Faisal Alkhateeb, Bilal H. Abed alguni and Mohammad Hani Al-rousan. Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem. The Journal of Supercomputing, 78: 4799–4826, 2022.

[16] Carlos Cotta Ferrante Neri. Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation, 2: 1–14, 2012.

[17] Lingaraj Haldurai. A study on genetic algorithm and its applications. International Journal of Computer Sciences and Engineering, 10: 139– 143, 2016.

[18] Goar, D. V. . (2021). Biometric Image Analysis in Enhancing Security Based on Cloud IOT Module in Classification Using Deep Learning- Techniques. Research Journal of Computer Systems and Engineering, 2(1), 01:05. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/9

[19] Indah Soesanti Ica Kurnia Hildayanti and Adhistya Erna Permanasari. Performance comparison of genetic algorithm operator combinations for optimization problems. In International Seminar on Research of Information Technology and Intelligent Systems, pages 43–48, 2018.

[20] Kathryn A. Dowsland, Jonathan M. Thompson. Simulated annealing. pages 1623–1655, 2012.

[21] K.Jayavani, G.M.Kadhar Nawaz. Study of genetic algorithm, an evolutionary approach. International Journal on Recent and Innovation Trends in Computing and Communication, 2: 2331–2334, 2014.

[22] Ólafur, J., Virtanen, M., Vries, J. de, Müller, T., & Müller, D. Data-Driven Decision Making in Engineering Management: A Machine Learning Framework. Kuwait Journal of Machine Learning, 1(1). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/108

[23] M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. Evolutionary Computation, 12: 273–302, 2004.

[24] Manju Sharma, Girdhar Gopal. Hill climbing based hybrid crossover in genetic algorithms. International Journal of Advanced Research in Computer Science and Software Engineering, 3: 468–473, 2013.

[25] Mhd. Furqan, Hartono, Erianto Ongko, Muhammad Ikhsan. Performance of arithmetic crossover and heuristic crossover in genetic algorithm based on alpha parameter. IOSR Journal of Computer Engineering, 19.

[26] Nitasha Soni, Tapas Kumar. Study of various crossover operators in genetic algorithms. International Journal of Computer Science and Information Technologies, 5: 7235–7238, 2014.

[27] Padmavathi Kora, Priyanka Yadlapalli. Crossover operators in genetic algorithms: A review. International Journal of Computer Applications, 162:34–36, 2017.

[28] Abdul Rahman, Artificial Intelligence in Drug Discovery and Personalized Medicine , Machine Learning Applications Conference Proceedings, Vol 1 2021.

[29] Petr Stodola, Karel Michenka, Jan Nohel, Marian Rybanský. Hybrid algorithm based on ant colony optimization and simulated annealing applied to the dynamic traveling salesman problem. Entropy, 22, 2020.

[30] S. Muthuraman, V. P. Venkatesan. A comprehensive study on hybrid meta-heuristic approaches used for solving combinatorial optimization problems. In 2017 World Congress on Computing and Communication Technologies (WCCCT), pages 185–190, 2017.

[31] Suresh, K. S. ., & Kamalakannan, T. . (2023). Digital Image Steganography in the Spatial Domain Using Block-Chain Technology to Provide Double-Layered Protection to Confidential Data Without Transferring the Stego-Object. International Journal of Intelligent Systems and Applications in Engineering, 11(2s), 61–68. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2508

[32] Santosh Kumar Suman, Vinod Kumar Giri. Genetic algorithms: Basic concepts and real world applications. International Journal of Electrical, Electronics and Computer Systems (IJEECS), 3:116–123, 2015.

[33] T O Ting, Xin-She Yang, Shi Cheng and Kaizhu Huang. Hybrid meta-heuristic algorithms: Past, present, and future. Recent Advances in Swarm Intelligence and Evolutionary Computation, 585:71–83, 2015.

[34] Yew Soon Ong, Natalio Krasnogor, Hisao Ishibuchi. Special issue on memetic algorithms. IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society, 37:2–5, 2007.

[35] Yılmaz Kaya, Murat Uyar, Ramazan Tekdn. A novel crossover operator for genetic algorithms: ring crossover. Neural and Evolutionary Computing), 2011.

[36] Ilhan Ilhan, An improved simulated annealing algorithm with crossover operator for capacitated vehicle routing problem. Swarm and Evolutionary Computation, 64, 2021.