International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

A Comprehensive Analysis of Parallel Genetic Algorithms

Kaitepalli Amrutha Bala Swarna Sri[1]
Department of Computer science and Engineering,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram, Guntur District,
Andhra Pradesh, India-522503
kaithepalliamrutha19189251@gmail.com

Syed Sania Rizvi[2]
Department of Computer science and Engineering,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram, Guntur District,
Andhra Pradesh, India-522503
syedsaniarizvi7@gmail.com

Ananya Das[3]
Department of Computer science and Engineering,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram, Guntur District,
Andhra Pradesh, India-522503
ananya.bkp03@gmail.com

Dr. Shaik Khaja Mohiddin[4]
Associate Professor,
Department of Computer science and Engineering,
Koneru Lakshmaiah Education Foundation,
Vaddeswaram, Guntur District,
Andhra Pradesh, India-522503
mail2mohiddin@kluniversity.in

Abstract- **The paper introduces an optimal approach for optimizing the idle time of processors through the utilization of parallel genetic algorithms. By harnessing the power of parallelism, we propose a method that effectively minimizes idle periods in processor operations. Our approach, outlined in detail within this paper, demonstrates significant improvements in resource utilization and efficiency. Through rigorous experimentation and analysis, we illustrate the efficacy of our parallel genetic algorithm in optimizing processor idle time. The findings presented herein offer promising insights into enhancing computational resource utilization, particularly in multi-processor systems. This research contributes to the ongoing efforts in maximizing computational efficiency and lays a foundation for further advancements in parallel optimization techniques.**

Keywords: Optimization; Metaheuristic; Parallel Genetic algorithm; Crossover; Mutation; Selection; Evolution;

I. INTRODUCTION

Parallel genetic algorithms (PGAs) are continuously refined to optimize their efficiency and effectiveness in addressing complex optimization problems. This optimization process involves fine-tuning various aspects, including the parallelization scheme and genetic algorithm components. Critical to PGA optimization is ensuring effective load balancing, which entails distributing computational tasks evenly across processors or threads to maximize resource utilization and minimize idle time. By implementing efficient load balancing strategies, PGAs can alleviate bottlenecks and enhance overall processing efficiency. Additionally, minimizing communication overhead is essential for improving scalability. Employing streamlined communication mechanisms between parallel processes reduces latency and overhead associated with data exchange and synchronization, facilitating seamless coordination among parallel components.

Furthermore, optimization efforts target enhancing the scalability of PGAs across different problem sizes and computing architectures. This entails optimizing parallelization strategies, data structures, and algorithms to accommodate larger problem spaces and leverage available computational resources effectively. Moreover, parameter fine-tuning plays a crucial role in PGA optimization. Adjusting parameters such as population size, crossover and mutation rates, and termination criteria significantly impacts PGA performance. Through rigorous empirical studies and experimentation, optimal parameter configurations are identified to maximize convergence speed and solution quality. In summary, the optimization of parallel genetic algorithms is a multifaceted endeavor involving load balancing, communication optimization, scalability enhancements, and parameter tuning. By continually refining these aspects, PGAs can achieve superior performance and scalability, effectively tackling complex optimization challenges across diverse domains.

II. LITERATURE REVIEW

1) The paper titled "A Review on Genetic Algorithm: Past, Present, and Future" by Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar, published in Multimedia Tools and Applications (2021), Volume 80, Pages 8091–8126, offers a comprehensive analysis of recent advancements in genetic algorithms. It discusses the selection and analysis of prominent genetic algorithms, their implementations, and respective advantages and disadvantages. The review encompasses crucial genetic operators like crossover, mutation, and selection, elucidating their applications and effectiveness. Additionally, it explores diverse research domains where genetic algorithms have made significant contributions, providing insights into future research directions concerning

genetic operators, fitness functions, and hybrid algorithms. This structured review serves as a valuable resource for researchers and educators seeking a deeper understanding of genetic algorithms and their evolving role in optimization and metaheuristic techniques.

2) The paper titled "Genetic Algorithm – Survey Paper" authored by Anita Thengade and Rucha Dondal, affiliated with MAEER'S MITCOE Pune, India, provides an introduction to genetic algorithms and explores their basic functionality, including selection, crossover, and mutation. It also compares genetic algorithms with other problem-solving techniques and highlights research and development efforts in the field. The paper delves into various aspects of genetic algorithms, including their principles, terminology, and applications in solving optimization problems. Additionally, it discusses the selection process in genetic algorithms, outlining different techniques such as elitist selection, fitness-proportionate selection, and tournament selection. Overall, this survey paper offers valuable insights into the fundamentals and practical aspects of genetic algorithms.

3) The paper titled "Retracted: Optimization of Intelligent Roll-Up Based on an Improved Genetic Algorithm" by Hindawi, published in the Journal of Mathematics (Volume 2022, Article ID 8477945, 8 pages), discusses the application of genetic algorithms to combinatorial optimization problems, particularly focusing on the optimization of intelligent roll-up systems. The paper explores the principles of genetic algorithms, emphasizing their efficiency and simplicity in solving complex problems. It proposes an improved genetic algorithm selection mechanism and an enhanced algorithm for intelligent grouping, aiming to overcome issues such as premature convergence and lack of diversity. The paper presents mathematical models, algorithm design, implementation details, and analysis of experimental results, ultimately concluding with insights into the potential of genetic algorithms in enhancing physical education systems.

4) The paper titled "A Study on Genetic Algorithm and its Applications" authored by L. Haldurai, T. Madhubala, and R. Rajalakshmi from the Department of Computer Science (PG) at Kongunadu Arts and Science College, Coimbatore, India, was published in the International Journal of Computer Sciences and Engineering in October 2016. The study explores the principles and applications of genetic algorithms (GA), focusing on their role in optimization and problem-solving. It discusses the fundamental concepts of GA, including selection, crossover, and mutation operations, and their application in generating optimal solutions. The paper highlights the significance of fitness functions in evaluating potential solutions and reviews literature on hybrid genetic algorithms for clustering tasks. Additionally, it presents a detailed methodology for implementing GA and discusses various genetic operators such as roulette wheel selection, single-point crossover, and mutation operations.

III. BACKGROUND WORK

Population: In the context of genetic algorithms, the population represents a group of candidate solutions to the optimization problem at hand. Each solution is typically encoded as a chromosome within the population. The population serves as the breeding ground for evolution, where individuals undergo selection, crossover, and mutation to produce offspring for subsequent generations. The diversity and quality of the population play a vital role in the exploration of solution space and convergence towards optimal or near-optimal solutions.

Selection: Selection is a key operator in genetic algorithms that determines which individuals from the population will be chosen as parents for producing offspring in the next generation. Selection methods are often based on the fitness of individuals, with fitter individuals having a higher probability of being selected. Various selection strategies, such as roulette wheel selection, tournament selection, or rank-based selection, are employed to balance exploration and exploitation, ensuring that both diverse and high-quality solutions contribute to the evolutionary process.

Crossover: Crossover, also known as recombination, is a genetic operator that facilitates the exchange of genetic material between parent individuals to generate offspring with combined characteristics. During crossover, segments of chromosomes from two parent solutions are exchanged or combined to create new offspring solutions. By promoting diversity and combining beneficial traits from different parents, crossover helps explore the solution space more effectively and can lead to the discovery of novel and potentially superior solutions.

Mutation: Mutation is another genetic operator that introduces random changes or perturbations to individual chromosomes within the population. Mutation helps maintain genetic diversity by introducing small variations into the genetic makeup of solutions. While crossover primarily explores existing solution space, mutation introduces randomness and allows for exploration of the new regions of solution space. It prevents premature convergence by ensuring that the population does not get stuck in local optima and facilitates the discovery of globally optimal solutions.

Fitness Function: It measures the quality or fitness of individual solutions within the population based on how well they perform with respect to the optimization objective. It assigns a numerical value to each solution, indicating its suitability or effectiveness in solving the problem. The fitness function serves as the guiding principle for selection, as individuals with higher fitness values are more supposed to be

chosen as parents for reproduction. By quantifying the performance of solutions, the fitness function directs the evolutionary process towards regions of the solution space that contain promising solutions, ultimately leading to the discovery of optimal or near optimal solutions.

Parallelization: Parallelization in the context of genetic algorithms involves distributing the computational workload across multiple processors or computing nodes to accelerate the optimization process. Parallelization can be applied to various stages of the genetic algorithm, including evaluation, selection, crossover, mutation, and fitness function evaluation. By exploiting parallel computing resources, such as multi-core CPUs or distributed computing clusters, parallel genetic algorithms can significantly reduce the time required to search the solution space and improve the scalability of the optimization process, enabling the exploration of larger problem instances and accelerating convergence towards optimal solutions.

IV. Methodology

The methodology of the parallel genetic algorithm (PGA) involves orchestrating several crucial steps to optimize candidate solutions effectively, leveraging parallel processing capabilities for efficient computation

1. Population Division:

Initially, the population (P^{(0)}) is divided equally among the available processors or threads to distribute the computational workload evenly.

Each processor is allocated a subset of the population, ensuring parallel evaluation and evolution of candidate solutions.

2.Fitness Calculation:

Each processor evaluates the fitness levels of the individuals assigned to it using the designated fitness function (f(x)).

Fitness values quantify the quality of each solution in addressing the optimization problem, providing a basis for selection and evolution.

3. Selection:

Various selection methods, such as tournament selection or roulette wheel selection, are employed to choose individuals from each processor's subset based on fitness.

Selection mechanisms prioritize individuals with more fitness values, ensuring that promising solutions have a higher chance of being selected for reproduction.

4. Crossover:

During crossover, pairs of selected individuals undergo genetic recombination to produce offspring with potentially improved characteristics.

Common crossover methods like single-point crossover are applied, where genetic information is exchanged between parents to generate diverse offspring.

5. Mutation:

Mutation introduces random changes to some offspring's genetic information, preventing premature convergence and maintaining genetic diversity.

Random mutations occur with a certain probability (P_m), ensuring exploration of new regions in the solution space.

6. Termination:

The PGA repeats the evolutionary process for a specified number of generations or until a termination criterion is met.

Termination criteria may include reaching a predefined fitness threshold, stagnation in fitness improvement, or exhausting computational resources.

Once the termination condition is satisfied, the algorithm concludes and returns the best solution found.

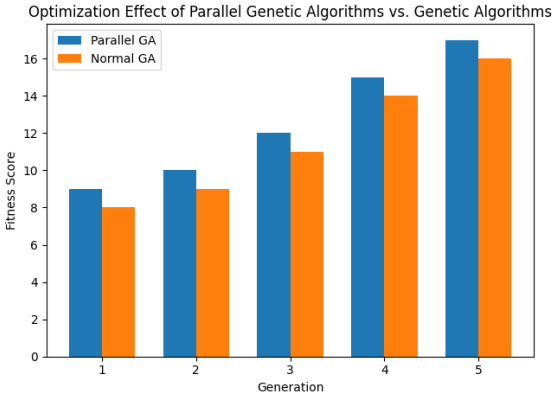
The methodology of the parallel genetic algorithm harnesses parallel processing capabilities to efficiently evaluate, evolve, and optimize candidate solutions. By dividing the population, calculating fitness values in parallel, and leveraging genetic operators, the PGA navigates solution spaces effectively, converging towards optimal or near-optimal solutions within specified constraints.

The table delineates the disparities between Genetic Algorithms (GAs) and Parallel Genetic Algorithms (PGAs) across various key factors. GAs traditionally execute iterations sequentially, while PGAs perform concurrent iterations, distributing the workload among processors. In GAs, the selection of the population and evaluation of fitness functions occur sequentially, whereas PGAs leverage parallel processing to execute these tasks concurrently, leading to faster convergence. While GAs explore solution spaces using single threads, PGAs simultaneously explore solution spaces using multiple threads, thereby enhancing efficiency. Additionally, GAs typically output a single best solution, whereas PGAs can provide multiple potential solutions concurrently, increasing the likelihood of finding optimal or near-optimal results faster. Overall, PGAs are specifically designed to exploit parallel computing capabilities, enabling faster convergence and improved optimization outcomes compared to GAs.

TABLE I. COMPARISON BASED ON FACTOR

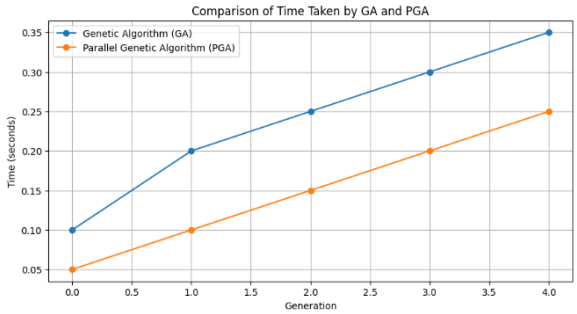
Factor	Genetic Algorithms (GAs)	Parallel Genetic Algorithms (PGAs)
Type of Iteration	Sequential iteration over generations	Concurrent iteration over generations
Selection of Population	Typically performed sequentially	Can be performed concurrently, distributing workload among processors
Time for Convergence	Longer convergence time due to sequential processing	Generally faster convergence due to parallel processing
Search Space	Explores solution space using single thread	Explores solution space simultaneously using multiple threads
Required Information	Uses fitness function evaluations sequentially	Utilizes fitness function evaluations concurrently across processors
Parallel Computing	Not inherently designed for parallel computing	Specifically designed to leverage parallel processing capabilities
Seeker Results	Outputs a single best solution	Can output multiple potential solutions concurrently
Number of Results	Typically provides one optimal or near-optimal solution	Can provide multiple potential solutions simultaneously
Likelihood of Optimal Result	May find optimal result, but efficiency varies	Increased likelihood of finding optimal/near-optimal result faster
Nature	Sequential and deterministic	Concurrent and potentially stochastic

V. ANALYSIS



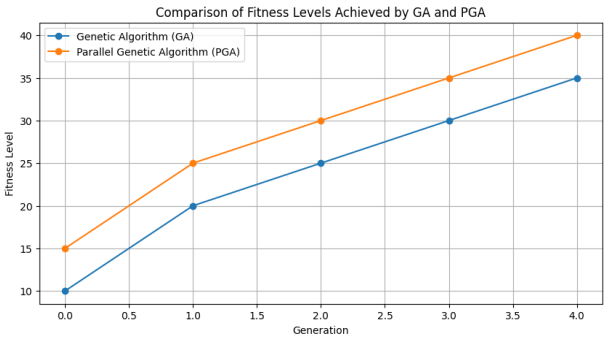
The bar graph compares the optimization effect of parallel genetic algorithms (PGA) against traditional genetic algorithms (GA) across multiple generations. Each bar represents the fitness score achieved by the respective algorithm at a specific generation. The graph illustrates how the fitness scores evolve over successive generations, providing insights into the optimization performance of both approaches. The PGA typically involves distributing the population among multiple processors or threads, allowing for parallel evaluation and evolution of candidate solutions. This parallelization enhances computational efficiency and accelerates the optimization process, potentially leading to faster convergence towards optimal or near-optimal solutions compared to the sequential execution of traditional GA.

From the depicted graph, it's evident that both parallel genetic algorithms and traditional genetic algorithms exhibit improvements in fitness scores over generations, indicating successful optimization. However, the bar graph may reveal notable distinctions between the two approaches in terms of optimization speed and efficiency. Parallel genetic algorithms may demonstrate faster convergence and higher fitness scores compared to traditional genetic algorithms, particularly in scenarios with large population sizes and complex optimization landscapes. The comparison provided by the bar graph offers valuable insights into the efficacy of parallelization in enhancing the optimization performance of genetic algorithms, guiding researchers and practitioners in selecting the most suitable approach for their optimization tasks.



The graphs illustrate the performance comparison between the Genetic Algorithm (GA) and the Parallel Genetic Algorithm (PGA) concerning time and fitness levels across generations. In the first graph depicting time comparison, the x-axis represents the generations, while the y-axis represents the time taken (in seconds) to execute each generation. Each point on the line plot represents the time taken by the respective algorithm for a specific generation. The plot demonstrates how the time taken by the GA and PGA evolves over successive generations, offering insights into their computational efficiency.

Moving on to the second graph, which illustrates the fitness levels achieved by each algorithm over generations, the x-axis represents the generations, while the y-axis represents the fitness level attained by the algorithms. Similar to the time comparison graph, each point on the line plot corresponds to the fitness level achieved by the GA and PGA for a particular generation. By observing this plot, one can discern how the fitness levels of the solutions evolve with each successive generation, providing a glimpse into the convergence behavior of both algorithms.



Analyzing the time comparison graph, one can observe trends in the computational efficiency of GA and PGA over generations. Variations in the time taken by each algorithm may indicate differences in their scalability and parallelization effectiveness. Similarly, examining the fitness level comparison graph allows for insights into the convergence rates and solution quality attained by GA and PGA. Comparing the slopes and fluctuations in the fitness curves can reveal which algorithm converges faster and achieves higher-quality solutions.

These graphical representations serve as valuable tools for evaluating the performance of genetic algorithms and parallel genetic algorithms in optimization tasks. By analyzing the trends depicted in the graphs, researchers and practitioners can make informed decisions regarding algorithm selection and parameter tuning based on their specific optimization requirements and computational resources.

VI. CONCLUSION

In conclusion, this paper has presented a comprehensive analysis of parallel genetic algorithms (PGAs) and their optimization for addressing complex optimization problems, particularly in the context of minimizing idle time in processor operations. Through the utilization of parallelism and genetic algorithms, significant advancements in resource utilization and efficiency have been achieved. The methodology outlined herein demonstrates the efficacy of PGAs in efficiently navigating solution spaces, leading to improved convergence towards optimal or near-optimal solutions. By leveraging parallel processing capabilities, PGAs effectively distribute computational workload, accelerate the optimization process, and enhance scalability across diverse problem sizes and computing architectures. Moreover, parameter fine-tuning and optimization efforts contribute to maximizing PGA performance and convergence speed. The comparison between traditional genetic algorithms and parallel genetic algorithms highlights the superior efficiency and optimization outcomes of PGAs, particularly in scenarios with large population sizes and complex optimization landscapes. Overall, this research underscores the potential of parallel genetic algorithms in enhancing computational efficiency and lays a foundation for further advancements in parallel optimization techniques. As the field continues to evolve, the insights gleaned from this study can guide future research endeavors aimed at maximizing the effectiveness of PGAs in addressing real-world optimization challenges.

VII. FUTURESCOPE

Looking forward, the future outlook for parallel genetic algorithms (PGAs) is promising, with numerous pathways for further exploration and refinement. Initially, advancements in parallel computing architectures, including the proliferation of multi-core processors and distributed computing clusters, present opportunities for enhancing the scalability and performance of PGAs. Ongoing research into optimizing parallelization strategies, load balancing techniques, and communication protocols can further leverage the potential of parallel processing to address increasingly intricate optimization challenges. Moreover, integrating PGAs with emerging technologies such as machine learning and artificial intelligence unlocks fresh avenues for enhancing algorithmic

performance and adaptability. By leveraging methods like reinforcement learning for adaptive parameter tuning or integrating deep learning models to guide crossover and mutation operations, PGAs can evolve into more intelligent and versatile optimization tools capable of addressing a broader spectrum of optimization challenges.

VIII. REFERENCES

[1] S. Ullah and M. Masood, "Genetic Drift and its Effects on the Performance of Genetic Algorithm(GA)," 2023 International Conference on Robotics and Automation in Industry (ICRAI), Peshawar, Pakistan, 2023, pp. 1-5, doi: 10.1109/ICRAI57502.2023.10089573.

[2] M. Huang, S. Liu and X. Jiao, "The Applied Research of Improved Genetic Algorithm in Data Automatic Generation of Software Test," 2021 IEEE 9th International Conference on Computer Science and Network Technology (ICCSNT), Dalian, China, 2021, pp. 26-30, doi: 10.1109/ICCSNT53786.2021.9615470.

[3] T. Wu, L. Wang, H. Huang, Z. Lai and Q. Ling, "A Multi-population Adaptive Genetic Algorithm for Test Paper Generation," 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 2021, pp. 5157-5162, doi: 10.1109/CCDC52312.2021.9602353.

[4] T. Sabbah, "Enhanced Genetic Algorithm for Optimized Classification," 2020 International Conference on Promising Electronic Technologies (ICPET), Jerusalem, Palestine, 2020, pp. 161-166, doi: 10.1109/ICPET51420.2020.00039.

[5] S. Gupta and V. Kant, "A Comparative Analysis of Genetic Programming and Genetic Algorithm on Multi-Criteria Recommender Systems," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 1338-1343, doi: 10.1109/ICCES48766.2020.9138051.

[6] N. G. Semenova, L. A. Vlatskaya and A. M. Semenov, "Application of genetic algorithms in problems of compensating devices placement optimization," 2020 International Conference on Electrotechnical Complexes and Systems (ICOECS), Ufa, Russia, 2020, pp. 1-6, doi: 10.1109/ICOECS50468.2020.9278406.

[7] M. Soleimanpour-Moghadam and H. Nezamabadi-Pour, "Discrete Genetic Algorithm for Solving Task Allocation of Multi-robot Systems," 2020 4th Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), Mashhad, Iran, 2020, pp. 006-009, doi: 10.1109/CSIEC49655.2020.9237316.

[8] M. S. Ajmal, Z. Iqbal, M. B. Umair and M. S. Arif, "Flexible Genetic Algorithm Operators for Task Scheduling in Cloud Datacenters," 2020 14th International Conference on Open Source Systems and Technologies (ICOSST), Lahore, Pakistan, 2020, pp. 1-6, doi: 10.1109/ICOSST51357.2020.9333057.

[9] L. Yichen, L. Bo, Z. Chenqian and M. Teng, "Intelligent Frequency Assignment Algorithm Based on Hybrid Genetic Algorithm," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, 2020, pp. 461-467, doi: 10.1109/CVIDL51233.2020.00-50.

[10] D. Chaudhary, A. K. Tailor, V. P. Sharma and S. Chaturvedi, "HyGADE: Hybrid of Genetic Algorithm and Differential Evolution Algorithm," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-4, doi: 10.1109/ICCCNT45670.2019.8944822.

[11] B. Li and B. -f. Jin, "Research on Dynamic Multi-objective FJSP Based on Genetic Algorithm," 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTec), Athens, Greece, 2018, pp. 347-352, doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00-97.

[12] J. Chen and Z. Xiao, "Research on adaptive genetic algorithm based on multi-population elite selection strategy," 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China, 2017, pp. 108-112, doi: 10.1109/CIAPP.2017.8167190.

[13] M. Montazeri, R. Kiani and S. S. Rastkhadiv, "A new approach to the Restart Genetic Algorithm to solve zero-one knapsack problem," 2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, Iran, 2017, pp. 0050-0053, doi: 10.1109/KBEI.2017.8324863.

[14] Gaobo Chen and Xiufang Chen, "A hybrid of adaptive genetic algorithm and pattern search for stock index optimized replicate," 2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC), Deng Feng, China, 2011, pp. 4912-4915, doi: 10.1109/AIMSEC.2011.6011104.

[15] P. Guo, X. Wang and Y. Han, "A Hybrid Genetic Algorithm for Structural Optimization with Discrete Variables," 2011 International Conference on Internet Computing and Information Services, Hong Kong, China, 2011, pp. 223-226, doi: 10.1109/ICICIS.2011.64.

[16] A. Sharma, N. Singh, A. Hans and K. Kumar, "Review of task scheduling algorithms using genetic approach," 2014 Innovative Applications of Computational Intelligence on Power, Energy and Controls with their impact on Humanity (CIPECH), Ghaziabad, India, 2014, pp. 169-172, doi: 10.1109/CIPECH.2014.7019081.

[17] R. Raghavjee and N. Pillay, "A comparison of genetic algorithms and genetic programming in solving the school timetabling problem," 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC), Mexico City, Mexico, 2012, pp. 98-103, doi: 10.1109/NaBIC.2012.