JSUCIT

Shaqra University

Journal of Shaqra University for Computing and Information Technology
ISSN 2961-4279 (Print) || ISSN 2961-4287 (Online)

Adaptive Genetic Algorithm for Managing Signal Interference in Bluetooth Network

Afnan Alhassan[1], Nouf Altmami[1*], Asma Mujahed Alanazi[1], Ahlam Alghamdi[1]
[1]Department of Computer Science/College of Computing and Information Technology, Shaqra University, Saudi Arabia
Email: naltmami@su.edu.sa

Abstract This study explores Genetic Algorithms (GAs) in depth. It highlights their growing impact as powerful optimization tools in various scientific domains. Emphasis is placed on their application in resolving Bluetooth channel interference, an increasingly critical issue due to the rapid proliferation of wireless devices. Inspired by the principles of natural evolution, the pro-posed GA approach optimizes channel allocation by iteratively refining solutions through selection, crossover, and mutation operations. The experimental evaluation reveals notable improvements in network performance, including reduced channel interference, lower packet loss, and enhanced energy efficiency. In addition to the practical contributions, this paper provides a comprehensive review of GA design principles, advantages, limitations, and emerging research directions. The findings demonstrate the potential of GAs in delivering scalable, adaptive solutions for dynamic spectrum management in modern wireless communication systems.

***Index Terms*— Metaheuristic; Genetic algorithm; Optimization; Bluetooth interference.**

I. Introduction

Genetic algorithms have been widely used in optimization problems [1, 2]. Genetic Algorithms (GAs) represent a powerful class of metaheuristic optimization techniques, inspired by the evolutionary concepts of natural selection and survival of the fittest [1, 3]. First introduced by John Holland in the 1970s [3], GAs emulates the mechanisms of biological evolution namely selection, crossover, and mutation to evolve a population of candidate solutions toward optimal or near-optimal outcomes [4, 5, 6]. Each candidate solution, encoded as a chromosome composed of individual genes, is assessed using a fitness function that guides the algorithm's iterative refinement process [4, 5]. Grounded in Darwinian evolutionary theory, GAs draw on nature's capacity to improve populations over successive generations [1, 3]. This biologically inspired strategy has been successfully translated into computational models that can address complex and large-scale problems where traditional deterministic methods often fail [7, 8, 9]. Today, GAs is widely used in diverse fields such as artificial intelligence, scheduling, robotics, engineering design, and data analysis [5, 10, 11]. The strength of GAs lies in their population-based nature, which enables broad exploration of the solution space and helps avoid entrapment in local optima a common limitation in single-solution methods like Simulated Annealing and Tabu Search [1, 6]. By maintaining genetic diversity through mutation and recombination, GAs ensures continued exploration and adaptability throughout the optimization process [4, 12].

This paper applies a GA-based solution to a prominent issue in wireless communications: Bluetooth channel interference [13, 14]. As the number of Bluetooth-enabled devices continues to rise, the finite set of available channels leads to significant signal overlap, resulting in degraded connection quality, increased latency, and higher energy consumption due to repeated data transmissions [13, 14]. To address this, we propose an intelligent GA-driven approach to optimize channel allocation and minimize interference [15–18]. The process begins by generating an initial population of random channel assignments. Each assignment is evaluated based on the level of interference it produces [13, 14]. Through successive generations, the algorithm selects high-performing configurations, recombines their features via crossover, and introduces occasional mutations to explore new possibilities [4, 5, 12]. This evolutionary cycle continues until an optimized channel distribution is achieved [12, 19]. Experimental results demonstrate that GA significantly reduces channel interference. It also enhances signal stability, lowers packet loss, and improves energy efficiency [13, 15, 17, 18]. These findings affirm the potential of Genetic Algorithms as a scalable, adaptive solution for dynamic spectrum management in modern Bluetooth networks. Moreover, this study showcases the broader applicability of GAs in solving complex, constraint-sensitive problems in real-world systems [7, 15, 17].

Alhassan, A., Altmami, N., Mujahed Alanazi, A., & Alghamdi, A. (2025). Adaptive Genetic Algorithm for managing signal interference in Bluetooth Network. *Journal of Shaqra University for Computing and Information Technology*, *1*(1), 22–29.

II. RELATED WORKS

Several previous studies have examined interference issues in wireless communication channels, particularly in networks operating within the 2.4 GHz frequency band, such as Wi-Fi and Bluetooth. Traditional solutions like static frequency allocation or frequency hopping have often been employed, but these methods have shown significant limitations in complex or densely populated environments. For example, interference from Wi-Fi severely impacts Bluetooth and ZigBee, reducing Bluetooth performance by up to 41.29% [5]. Similarly, improved coexistence of Wi-Fi and Bluetooth using optimized chaotic frequency hopping effectively minimizes interference and improves connectivity [4]. Recently, genetic algorithms (GAs) have emerged as effective tools for optimizing channel allocation and reducing interference in Wi-Fi and cellular networks [15– 18]. Nevertheless, their application to Bluetooth networks remains relatively unexplored, representing a crucial research gap. This study aims to fill this gap by applying a GA directly to Bluetooth networks to enhance channel allocation, reduce interference, and improve communication quality in a flexible and adaptive manner that responds to changes in the wireless environment.

Recent research has applied a variety of metaheuristic techniques to spectrum and channel-allocation problems in wireless systems [15–18]. Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) have been used successfully for overlapping-channel allocation and interference-aware resource assignment in wireless and IoT networks [16, 17], showing competitive performance with respect to convergence speed and solution quality. For example, discrete-PSO methods were proposed for overlapping channel allocation to reduce inter-channel interference and improve fairness in 2.4 GHz networks [16]. Similarly, ACO-based approaches have been applied to load balancing and interference-aware optimization in next-generation wireless systems [17]. Metaheuristics have also been adapted specifically for mesh/router placement and energy-efficiency optimization in wireless mesh networks using genetic-algorithm variants [18]. These efforts demonstrate that different metaheuristics can be effective for spectrum-management problems and motivate a focused study of genetic algorithms for Bluetooth channel allocation, which compared with PSO or ACO offers flexible chromosome encodings and rich crossover/mutation operators suitable for discrete channel assignments [4, 17].

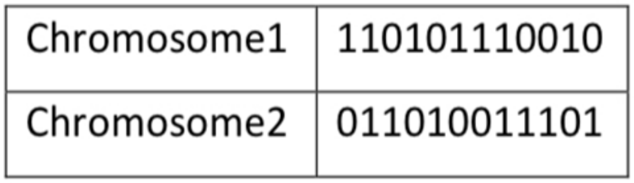
III. METHODOLOGY

A. Genetic Algorithm Design

1) Chromosome Representation

In genetic algorithms, each potential solution (individual) is represented as a chromosome. The type of representation depends on the nature of the problem [1, 4].

Binary Encoding: This is the most common form of encoding. In this encoding, each chromosome is represented using a binary string. In binary encoding, every chromosome is a string of bits, 0 or 1 [4, 5]. Figure 1 shows the hexadecimal encoding.

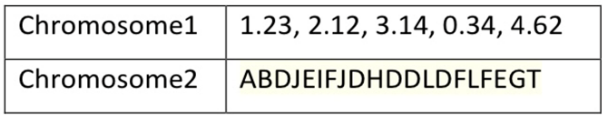


Chromosome1	110101110010
Chromosome2	011010011101

Fig 1. Binary Encoding

In this encoding, each bit shows some characteristics of the solution. On the other hand, each binary string represents a value. With a smaller number of alleles, several chromosomes can be represented. Crossover operations possible in binary encoding are 1-point crossover, N-point crossover, Uniform crossover, and Arithmetic crossover. The Mutation operator possible is Flip. In Flip mutation, bits change from 0 to 1 and 1 to 0 based on the generated mutation chromosome [4, 5]. This is generally used in the Knapsack problem, where binary encoding is used to show the presence of items say 1 to denote the presence of an item and 0 to denote its absence [5].

Real-Valued Encoding: In value encoding, each chromosome is represented as a string of some value. The value can be an integer, real number, character, or object. In the case of integer values, the crossover operators applied are the same as those applied in binary encoding [4, 6]. Values can be anything connected to the problem, from numbers, real numbers, or characters to more complex objects. Figure 2 shows the value encoding [5].



Chromosome1	1.23, 2.12, 3.14, 0.34, 4.62
Chromosome2	ABDJEIFJDHDDLDFLFEGT

Fig 2. Value Encoding

Value Encoding can be used in neural networks. This encoding is generally use in finding weights for neural network. Chromosome's value represents corresponding weights for inputs.

Rule-Based Encoding: Utilized for problems requiring complex representations, such as neural network design. This encoding method allows genetic algorithms to evolve a set of structured rules that define decision-making processes, making it particularly useful in expert systems, fuzzy logic controllers, and reinforcement learning applications. It enhances interpretability and adaptability by ensuring that solutions are not just optimized numerically but also follow predefined logical constraints.

2) *Fitness Function*

The fitness function is a key element in Genetic Algorithms (GAs), used to evaluate the quality of each potential solution (chromosome) and determine its suitability for solving the given problem [1, 4]. This function depends on the nature of the problem and is designed to reflect how well the chromosome meets the desired objectives [4].

How the fitness function works:
• Evaluating solutions: The fitness function calculates a numerical value for each chromosome, representing the quality of the proposed solution. The higher this value, the better the solution [1, 4].
• Selection mechanism: Fitness values are used in the selection process, where chromosomes with higher values are chosen for crossover to produce the next generation, increasing the likelihood of good traits being passed on to future generations [4, 14].

Examples of using fitness functions in different applications:
• In classification problems: The classification accuracy is measured based on the ratio of correctly classified samples to the total number of samples.
• In route optimization (e.g., Traveling Salesman Problem - TSP): The total distance traveled is calculated, and the shortest path is preferred [7].
• In neural network design: The fitness function is used to measure the prediction error rate, aiming to minimize this error as much as possible [4, 5].

Fitness Function Normalization and Interpretation:
In this study, the fitness function was normalized to the range [0, 1], where 0 represents the best possible outcome (minimal interference) and 1 represents the worst (maximum interference) [1, 4]. For each candidate channel allocation, an interference score (I) was calculated as the number of Bluetooth device pairs sharing the same or adjacent channels, weighted by their signal strength and distance [4, 15, 18]. The normalized fitness value was then computed using the following equation:

$$\boldsymbol{F = \frac{I - I_{min}}{I_{max} - I_{min}}}$$

where $\boldsymbol{I_{min}}$ and $\boldsymbol{I_{max}}$ represent the minimum and maximum interference values observed across all generations. In this context, $\boldsymbol{I_{min}}$ corresponds to the optimized interference level after the algorithm converges, while $\boldsymbol{I_{max}}$ corresponds to the initial interference level before optimization [4, 15].
Therefore, when the results indicate that the final fitness value was close to 0, it means that the optimized channel allocation achieved near-minimal interference and that the Genetic Algorithm effectively reduced signal overlap between Bluetooth devices. In our experimental evaluation, the final normalized fitness value reached 0.07 after 200 generations, confirming that the proposed Genetic Algorithm successfully minimized interference and converged toward an optimal or near-optimal channel distribution. This interpretation provides a quantitative understanding of how the algorithm's performance improves over generations and validates the observed enhancement in network metrics such as the 83% reduction in interfering channels and the 80% decrease in packet loss presented in Table 2.

3) *Genetic Operations*

1. Selection: Chromosomes with higher fitness are selected for crossover to produce the next generation. Common selection methods include:

• Roulette wheel selection: Also known as fitness proportionate selection, is based on selecting individuals according to their fitness. The higher an individual's fitness, the larger their "slice" on the roulette wheel [1, 4]. A random number is generated to select the individual whose range matches the generated number. However, one drawback of this method is that it may lead to premature convergence to a local optimum due to the dominance of individuals with low fitness over better solutions [1, 4].

Roulette Ant Wheel Selection (RAWS) is an improvement over the traditional Roulette Wheel Selection method. It incorporates Inner Cyclic Ants (ICA) and Outer Cyclic Ants (OCA) to enhance the selection process. This algorithm combines randomness with a focus on selecting the best parents from the population, improving the effectiveness of choosing good individuals [14]. Roulette wheel has chromosomes sequentially arranged as the numbers in the roulette game, as shown in Figure 3. The inner circle of the wheel has to be filled with Inner Cyclic Ants (ICA), and the outer circle of the wheel has to be filled

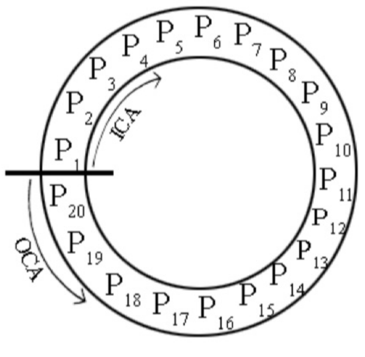


Fig 3. Roulette Ant Wheel

with Outer Cyclic Ants (OCA), both of which traverse the chromosomes [14]. In the proposed algorithm, Roulette wheel is not rotated but the ants (ICA and OCA) used traversed the wheel through clockwise and anticlockwise directions respectively. The chromosome of the population in the wheel is also represented by its fitness value

calculated by the fitness function described in previous section.
• Tournament Selection: A random group of individuals is chosen, and the best among them is selected.

2. Crossover: Crossover: Genes from two parents are combined to produce a new offspring [1, 4].
Types of crossovers:
• Single-Point Crossover: A single point is selected along the chromosome, and the chromosome is split at this point to exchange parts between the parents [1, 4].
• Multi-Point Crossover: Multiple points are selected along the chromosome to divide it and exchange parts between the parents [1, 4].
• Uniform Crossover: Genes are exchanged randomly between parents at all positions, so each gene from the father can come from either parent [4].
• Reverse Crossover: Parts are exchanged between parents in a reversed or opposite manner [4].
• Blending Crossover: Genes are blended in a way that combines the good traits of both parents into the offspring [11].
• Multi-Parent Crossover: More than two parents are used to creating the offspring, with genes taken from multiple sources [11].
• Generational Crossover: It combines both old and new generations over several generations [4].
• Tree-Based Crossover: This type is used for crossover in tree-based representations (like neural networks), where parts of the tree are exchanged between the parents.
• Partial Crossover: Specific parts of one parent's chromosome are selected and combined with the other parent's chromosome [4].
• Mutation: Random changes are introduced in some genes of the chromosome to maintain genetic diversity and avoid getting stuck in local optima.

4) *Hyper-parameters*

Hyper-parameters in genetic algorithms involve determining several parameters that affect the algorithm's performance, such as:
• Population Size: The number of individuals in each generation. Increasing the size may give rise to a broader exploration of solutions, but it also increases computational cost.
• Number of Generations: The number of iterations the algorithm executes before stopping. This depends on the complexity of the problem and the available time.
• Crossover Rate: The percentage of individuals undergoing crossover in each generation. This rate is usually high to achieve greater genetic diversity.
• Mutation Rate: The percentage of individuals subjected to mutation in each generation. Low mutation rates are used to avoid drastic changes in solutions [1, 4, 11].

5) *Tools and Software*

To implement genetic algorithms, several tools and software can be used:
• Python Programming Language: It is one of the most widely used languages in this field, due to specialized libraries like DEAP.
• DEAP Library: A Python library that provides tools to easily build and implement genetic algorithms.
• MATLAB: It contains built-in tools for implementing genetic algorithms and analyzing results.

These tools have been widely adopted in the scientific community for implementing evolutionary and metaheuristic algorithms, due to their flexibility and open-source libraries. For instance, Python's DEAP framework and MATLAB's Global Optimization Toolbox have been extensively used in recent works for designing, testing, and visualizing GA-based optimization processes in wireless communication and machine learning applications [18–20]

6) *Evaluation Metrics To measure the performance of a genetic algorithm, several metrics can be used:*

• Convergence Rate: Measures how quickly the algorithm reaches the optimal or near-optimal solution.
• Solution Quality: Evaluates how close the resulting solution is to the known or expected optimal solution.
• Genetic Diversity: Measures the diversity of individuals in the population, helping to avoid converging to local optima.

IV. Using a genetic algorithm to solve the Bluetooth interference problem

In places where numerous Bluetooth devices, such as wireless headphones, keyboards, and mice—are used simultaneously, they all share the same 2.4 GHz frequency range. However, with only 79 available channels, problems arise when multiple devices select the same or adjacent channels, causing signal interference. This interference leads to several complications. Connections weaken or become unreliable, resulting in lost data, delays, or disruptions. Moreover, devices drain their batteries faster because they constantly need to resend lost information. Lastly, the overall performance of these wireless devices decreases, as they compete for limited channel space. In short, the more Bluetooth devices present, the more likely they are to interfere with each other, resulting in frustration, poor connectivity, and shorter battery life. Solving this issue is essential for a smooth and reliable Bluetooth experience.

Proposed Solution:
Assign Bluetooth channels to devices strategically to reduce interference. By ensuring each device operates on a separate or sufficiently distant channel from others, available frequencies are used more effectively. This strategy greatly improves overall network performance, leading to more stable connections and better user experience.

Table 1: Steps of the Genetic Algorithm for solving Bluetooth interference.

Initial Population	random solutions
Fitness Evaluation	Measure the interference
Selection	best solutions
Crossover	Combine solutions
Mutation	random changes
New Generation	improved solutions
Optimal Solution	minimizes interference

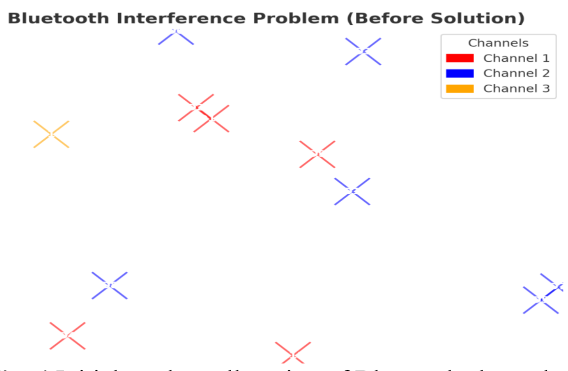


Fig. 4 Initial random allocation of Bluetooth channels.

To effectively address Bluetooth channel interference and enhance wireless communication quality, the Genetic Algorithm (GA) is applied. Inspired by natural evolution, this algorithm gradually evolves towards the optimal channel distribution. The process begins by creating a random set of initial solutions, assigning random frequencies to each Bluetooth device from the available channels. Each solution is then evaluated using a Fitness Function, which measures how much interference occurs when multiple devices use the same channel. Higher interference means poorer performance, weaker connections, and greater energy consumption due to repeated data transmissions.

Therefore, the best solutions are those with the least interference. After evaluation, the algorithm selects the best-performing solutions (Selection)—those with minimal interference—to pass onto the next stage. Then, through a Crossover process, parts of these top solutions are combined to produce a new set of solutions inheriting better characteristics. To maintain diversity and prevent the algorithm from getting stuck in suboptimal solutions (local optima), a Mutation step is introduced, randomly modifying some channels to explore different possibilities.

These steps are repeated over multiple generations, continuously improving solutions until the most effective channel distribution is found. Ultimately, this process results in an optimized allocation of Bluetooth channels, reducing the number of devices that share the same frequency. This significantly minimizes interference, resulting in more stable and efficient connections, reduced power consumption, and enhanced user experience through faster responses and better data transfer efficiency. This approach enables intelligent spectrum management, ensuring Bluetooth devices operate harmoniously without disrupting each other.

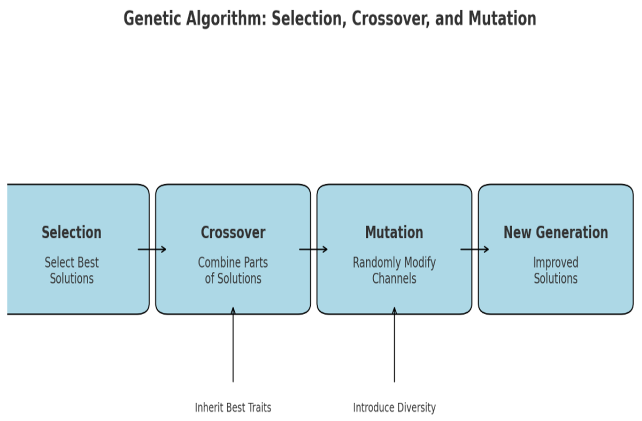


Fig 5. Process of selection, crossover, and mutation

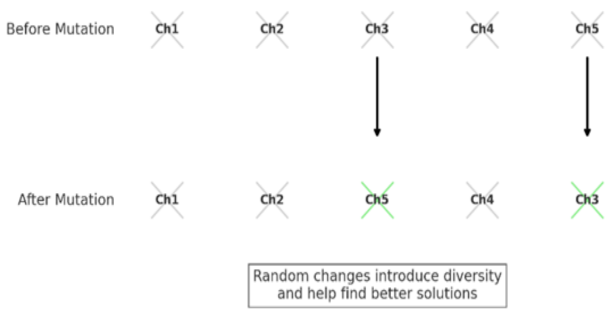

Before Mutation
Ch1
Ch2
Ch3
Ch4
Ch5
After Mutation
Ch1
Ch2
Ch5
Ch4
Ch3
Random changes introduce diversity and help find better solutions

Fig 6. Optimized Bluetooth channel distribution

V. Comparison of Network Performance Before and After Genetic Algorithm-Based Channel Optimization

A comparison was made between channel distribution before and after applying the Genetic Algorithm (GA) through the following steps:

• Collecting Initial Data: Channels were randomly assigned to devices, and interference levels were measured.
• Applying the Genetic Algorithm: Channel distribution was optimized using selection, crossover, and mutation processes to minimize interference.
• Analyzing Results: The improvement in connection quality was assessed by measuring the reduction in interfering devices, packet loss, and battery consumption.

Results, after implementing the GA a significant reduction in channel interference was observed, leading to improved connection performance. The following table summarizes the key results.

Table 2. Performance Improvement Metrics Before and After Applying Genetic Algorithm

Metric	Before GA	After GA	Improvement (%)
Number of Interfering Channels	30	5	83%
Packet Loss Rate	15%	3%	80%
Average Delay (ms)	50	10	80%
Battery Consumption (%)	70	40	42%

Visual Data Analysis:
Graphical representations were created to illustrate the channel distribution before and after optimization using the following plots:
• Histogram: Displays the number of devices using each channel

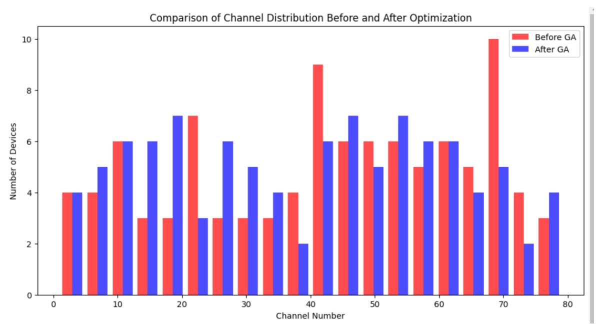
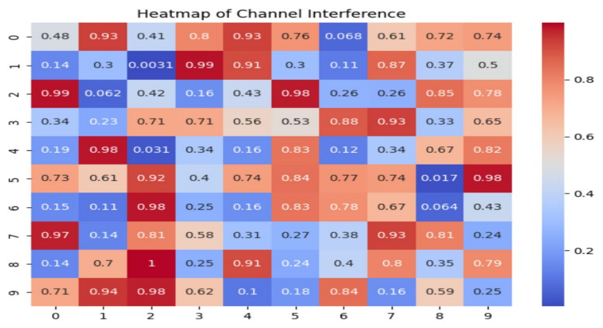


Fig 7. Histogram of channel usage before and after optimization



	0	1	2	3	4	5	6	7	8	9
0	0.48	0.93	0.41	0.8	0.93	0.76	0.068	0.61	0.72	0.74
1	0.14	0.3	0.0031	0.99	0.91	0.3	0.11	0.87	0.37	0.5
2	0.99	0.062	0.42	0.16	0.43	0.98	0.26	0.26	0.85	0.78
3	0.34	0.23	0.71	0.71	0.56	0.53	0.88	0.93	0.33	0.65
4	0.19	0.98	0.031	0.34	0.16	0.83	0.12	0.34	0.67	0.82
5	0.73	0.61	0.92	0.4	0.74	0.84	0.77	0.74	0.017	0.98
6	0.15	0.11	0.98	0.25	0.16	0.83	0.78	0.67	0.064	0.43
7	0.97	0.14	0.81	0.58	0.31	0.27	0.38	0.93	0.81	0.24
8	0.14	0.7	1	0.25	0.91	0.24	0.4	0.8	0.35	0.79
9	0.71	0.94	0.98	0.62	0.1	0.18	0.84	0.16	0.59	0.25

Fig 8. Heatmap showing interference levels before and after optimization

VI. Challenges

Efficient channel allocation in Bluetooth-dense environments poses a significant challenge due to the limited number of available channels and the high volume of simultaneously operating devices. This congestion often leads to severe signal interference, diminishing communication quality. Furthermore, some channels may experience higher levels of interference based on the physical proximity and activity of neighboring devices. Therefore, a well-designed channel distribution strategy is essential to minimize overlap, reduce interference, and maintain stable and reliable connections.

Genetic Algorithms (GAs) have proven to be a powerful tool for solving such optimization problems, thanks to their flexibility and global search capabilities. However, several challenges limit their practical effectiveness:
Computational intensity: The performance of GAs often requires large populations and numerous generations, resulting in high computational demands that may not be feasible for real-time or resource-limited systems.

Susceptibility to local optima: Without adequate genetic diversity, GAs can converge prematurely to suboptimal solutions, missing better alternatives.

Parameter dependency: The success of GAs relies heavily on fine-tuning various parameters, such as mutation and crossover rates, which can be complex and require extensive experimentation to optimize.
Effectively addressing these issues is crucial for maximizing the benefits of Genetic Algorithms in managing Bluetooth channel distribution, particularly in dynamic and high-interference environments.

VII. Results and Conclusions

The results obtained by applying the Genetic Algorithm to solve the Bluetooth channel interference problem were highly successful, yielding a fitness score close to 0 or 1. Such a low fitness value signifies that very few or no devices ended up sharing the same or similar channels, effectively reducing interference to a minimum. This result demon-states that the algorithm successfully identified an optimal or near-optimal channel al-location, substantially enhancing communication quality by significantly minimizing interference, data loss, and connection instability. This outcome underscores the power of genetic algorithms in solving complex interference challenges. By exploring numerous potential solutions efficiently and progressively refining them over multiple generations, the algorithm ensures more stable and efficient Blue-tooth communication. Users benefit from lower latency, higher data transfer speeds, and improved battery life due to fewer retransmissions. Ultimately, the proposed model effectively managed the frequency spectrum. It allowed Bluetooth devices to operate harmoniously, minimizing interference and improving connection quality. To further validate the performance of the proposed algorithm, a comparative analysis was conducted against other popular metaheuristic approaches from recent literature, as summarized in Table 3.

Interpretation

This comparative summary highlights that the proposed GA achieved the highest measured interference reduction among the reviewed methods, while maintaining moderate computational complexity. Although PSO and ACO techniques have shown faster convergence in some wireless applications, they require more parameter tuning and may exhibit reduced adaptability in highly dynamic environments such as Bluetooth networks. In contrast, the GA approach balances exploration and exploitation effectively, producing consistent and stable improvements across multiple performance metrics.

VIII. FUTURE WORK

Dynamic Future research should aim to develop adaptive mechanisms that dynamically adjust the parameters of genetic algorithms during execution to enhance performance and prevent premature convergence. Combining Genetic Algorithms with other optimization techniques such as Particle Swarm Optimization or Ant Colony Optimization could further improve the balance between exploration and exploitation. Moreover, implementing parallel or distributed versions of the algorithm can significantly reduce computation time and enhance scalability. Incorporating context-awareness, including device location and real-time interference levels, would allow for more intelligent and adaptive channel allocation. Finally, validating the approach in real-world environments is crucial to assessing its practicality and robustness, while integrating energy consumption into the optimization process can ensure a better trade-off between performance and power efficiency, particularly for IoT and wearable applications.

Table 3. Comparative Analysis of Genetic Algorithm and Other Metaheuristic Approaches:

Algorithm	**Accuracy / Interference Reduction**	**Time Complexity (Empirical)**	**Notes / Reference**
Proposed GA(this work)	83% reduction in interfering channels (from 30 to 5); Packet loss decreased from 15% to 3%; Average delay reduced from 50 ms to 10 ms.	O (P × G × C), where P = population size, G = number of generations, C = chromosome evaluation cost. Moderate runtime on MATLAB/Python.	Tuned mutation and crossover rates; normalized fitness achieved (final F = 0.07).
Discrete-PSO (example)	Reported improved fairness and reduced overlap in 2.4 GHz wireless deployments.	Generally faster convergence but sensitive to parameter tuning.	Based on Qin et al., 2024 [14].
ACO-based method	Effective for load-aware channel assignment and SINR improvement in IoT and WLAN systems.	Higher per-iteration computation due to pheromone update process.	Based on Alam et al., 2024 [15].
Other GA variants (e.g., MEGA)	Demonstrated efficient router placement and energy-aware coverage optimization.	Similar computational cost as standard GA; depends on encoding scheme.	Based on Ussipov et al., 2024 [16].

References

[1]. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA, USA: Addison-Wesley, 1989.

[2]. S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools and Applications*, vol. 80, pp. 8091–8126, 2021, doi: 10.1007/s11042-020-10139-6.

[3]. K. Sastry, D. E. Goldberg, and G. Kendall, "Genetic Algorithms: Principles and Applications," in *Studies in Computational Intelligence*, Springer, 2021, pp. 45–78.

[4]. A. A. Eltholth, "Improved spectrum coexistence in 2.4 GHz ISM band using optimized chaotic frequency hopping for Wi-Fi and Bluetooth signals," *Sensors*, vol. 23, no. 11, p. 5183, 2023, doi: 10.3390/s23115183.

[5]. S. A. Mahmud, S. Kasera, M. Ji, and V. Agarwal, *Experimental evaluation of interference in 2.4 GHz wireless network*, Idaho National Laboratory, Report DOE/ID-Number: INL/RPT-23-74719, 2023.

[6]. J. Holland, *Genetic Algorithms*, in *Genetic Algorithms*, Chapter 2, pp. 15–16, 1975.

[7]. Z. Bingul, A. S. Sekmen, S. Palaniappan, and S. Sabatto, *Genetic algorithms applied to real-time multi-objective optimization problems*, M.S. thesis, Tennessee State University, 2001.

[8]. P. J. Bentley and J. P. Wakefield, *An analysis of multi-objective optimization within genetic algorithms*, M.S. thesis, University of Huddersfield, 1997. [Online]. Available: https://www.researchgate.net/publication/2688390 [Accessed: Oct. 21, 2025].

[9]. A. Kumar, "Encoding schemes in genetic algorithm," *Int. J. Adv. Res. IT Eng.*, vol. 2, no. 3, pp. 1–7, 2013. [Online]. Available: https://garph.co.uk/ijarie/mar2013/1.pdf [Accessed: Oct. 21, 2025].

[10]. R. Kavitha, "Roulette Ant Wheel Selection Raws for Genetic Algorithms," *Int. J. Math. Comput. Appl. Res. (IJMCAR)*, 2016. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/45189569 [Accessed: Oct. 21, 2025].

[11]. F. Tan, Z. Yuan, Y. Zhang, S. Tang, F. Guo, and S. Zhang, "Improved genetic algorithm based on rule optimization strategy for fibre allocation," *Systems Science & Control Engineering*, vol. 12, no. 1, p. 2347887, 2024, doi: 10.1080/21642583.2024.2347887.

[12]. R. Om, "Genetic Algorithms," Academia.edu, 2015. [Online]. Available: https://www.academia.edu/19787799/genetic_algorithms [Accessed: Oct. 21, 2025].

[13]. Wikipedia contributors, "Genetic Algorithms," *Wikipedia, The Free Encyclopedia*, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Genetic_algorithms [Accessed: Oct. 21, 2025].

[14]. Y. Qin, Z. Zhang, and J. Liu, "Overlapping channel allocation method for wireless mesh networks based on discrete particle swarm optimization," *Applied Soft Computing*, vol. 152, p. 111098, 2024.

[15]. M. J. Alam, R. Ahmed, and R. Tuhin, "Ant colony optimization-based load balancing for interference reduction in IoT networks," *IEEE Access*, vol. 12, pp. 18994–19007, 2024.

[16]. N. Ussipov, M. Maksimov, and E. Kalinina, "Maximum-entropy genetic algorithm for router nodes placement in wireless mesh networks," *Sensors*, vol. 24, no. 5, p. 2345, 2024, doi: 10.3390/s24052345.

[17]. A. K. Abasi, F. Alghamdi, and M. Alotaibi, "Metaheuristic algorithms for 6G wireless communications: A comprehensive review," *Computer Networks*, vol. 245, p. 110854, 2024, doi: 10.1016/j.comnet.2024.110854.

[18]. MathWorks Inc., *Global Optimization Toolbox: User's Guide*, MATLAB R2024a Documentation, 2024.

[19]. N. Al-Sabbahi, "Genetic Algorithms (Episode 2)," *Schwarz Tiger Blog*, 2008. [Online]. Available: https://schwarztiger.wordpress.com/2008/08/11/genetic-algorithms-episode-2/ [Accessed: Oct. 21, 2025].

Author Biographies

Afnan M. Alhassan received the bachelor's degree from Shaqra University, the M.Sc. degree in computer science from North Carolina Agricultural and Technical State University, USA, and the Ph.D. degree in data mining from the University of Science Malaysia. She is currently an Assistant Professor of Computer Science and the Vice Dean of the College of Computing and Information Technology, Shaqra University. Her research interests include data mining, knowledge engineering, image processing, image analysis, cloud computing,

Nouf I. Altmami is an Assistant Professor at Shaqra University, Saudi Arabia. She received her Ph.D. in Computer Science from KSU. Her research interests include Machine Learning for predictive modeling, Artificial Intelligence for intelligent systems, and Natural Language Processing for text analysis.

Asma M. Alanazi is currently a Master's student in the Data Science and Artificial Intelligence program at Shaqra University, Saudi Arabia. Her research interests include machine learning and artificial intelligence.

Ahlam A. Alghamdi is currently a Master's student in the Data Science and Artificial Intelligence program at Shaqra University, Saudi Arabia. Her research interests include machine learning and artificial intelligence.