Optimizing Data Transmission and Energy Efficiency in Wireless Networks: A Comparative Study of GA, PSO, and Hybrid Approaches

Suhare Solaiman
Department of Computer Science-College of Computers and Information Technology, Taif University
P.O. Box 11099, Taif 21944, Saudi Arabia

***Abstract*—As wireless communication technology evolves, efficient resource allocation in Orthogonal Frequency Division Multiple Access (OFDMA) networks is becoming more important. This study looks at three resource allocation algorithms: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and a hybrid approach that combines both. The hybrid algorithm takes advantage of the strengths of both methods to improve data transmission and energy efficiency. Using simulations in MATLAB, the study assesses algorithms based on key metrics such as data rate, energy consumption, and computational complexity. The findings show that the hybrid approach generally performs better than both GA and PSO, especially in maximizing data rates. This research offers useful information for network operators looking to implement effective resource management strategies in practical wireless communication settings.**

***Keywords*—Resource allocation; optimization; genetic algorithms; particle swarm optimization; hybrid algorithm**

I. Introduction

As wireless communication technology rapidly evolves, the importance of efficient resource allocation has increased. Orthogonal Frequency Division Multiple Access (OFDMA) networks are the foundation of modern communication, enabling everything from our smartphones to high-speed Internet connections. OFDMA divides the available bandwidth into multiple distinct subcarriers, allocating a unique set to each user. This separation allows users to communicate simultaneously without interference, ensuring a seamless and efficient experience [1] [2].

OFDMA is commonly used in various wireless communication standards, including WiMAX, LTE, and 5G networks. It effectively manages bandwidth, allowing multiple users to connect at the same time while keeping latency low and throughput high. By dynamically allocating subcarriers based on user demand and channel conditions, OFDMA improves overall network performance. This makes it a practical choice for modern networks, where reliable connections for activities, such as video streaming and online gaming are increasingly important [3].

In a world where staying connected is essential, the task of effectively distributing limited resources, such as bandwidth and power, has become more complicated than ever. Energy efficiency plays an important role, helping to lower operational costs for network providers, extend battery life for mobile devices, and reduce the environmental impact of increased technology use. Optimizing resource allocation is increasingly seen as both a practical necessity and a responsible choice. Implementing energy-efficient practices can help reduce carbon footprints and support efforts to address climate change.

To address these challenges, a variety of algorithms are adopted, each offering unique strengths and tailored solutions for resource allocation. For instance, the Water-Filling Algorithm is recognized as a fundamental method for distributing power according to the varying channel conditions of different users. This algorithm efficiently allocates power to users with better channel quality, thereby maximizing overall system performance [4] [5]. In addition, the Bisection Algorithm plays a crucial role by systematically narrowing the search for optimal solutions, ensuring effective and efficient resource utilization. Maintaining high quality of service is especially important in crowded networks as the number of connected devices and bandwidth demands increase [6] [7].

Adaptive resource allocation challenges can also be addressed using heuristics such as Genetic Algorithm (GA) [8] and Particle Swarm Optimization (PSO) [9]. GA draws inspiration from natural processes, beginning with a set of potential solutions that are refined over time through selection, crossover, and mutation. As these solutions evolve across multiple generations, they continuously improve, making GA particularly effective for complex problems that traditional methods may struggle to solve [10] [11]. On the other hand, PSO mimics the collective behavior of birds and fish. In this approach, each *particle* represents a potential solution that navigates the solution space based on its own experiences and those of its neighbors. The position of a particle is adjusted according to two factors: the best solution it has discovered and the locations of its neighbors. This collaborative effort enables PSO to explore the solution space effectively, often leading to optimal solutions in complex scenarios [12] [13]. The hybrid approach combines the evolutionary characteristics of GA with the collaborative search capabilities of PSO, resulting in a more robust and efficient method for optimizing data transmission and energy efficiency in OFDMA networks [14] [15].

In this study, a comprehensive comparison of three resource allocation algorithms is presented: GA, PSO, and a hybrid method that effectively combines the strengths of both GA and PSO. The evaluation will focus on key performance metrics such as data rate, energy consumption, and time complexity. By analyzing these algorithms across a range of scenarios, the study aims to identify the best algorithm in terms of data rates and energy consumption. The findings seek to bridge the gap between theoretical models and practical applications, offering information that can enhance the operational efficiency of wireless networks. As the industry encounters increasingly complex and demanding environments, understanding how to optimize resource allocation will be essential for ensuring sustainable, high-quality service delivery. This research will contribute to informed decision making and strategic planning in resource management, ultimately supporting the evolving needs of modern communication systems.

A. Contributions

This work offers several key contributions to the field of wireless communication:

- A comparative evaluation of three existing algorithms, GA, PSO, and a hybrid algorithm that combines GA and PSO, is presented to optimize data transmission and energy efficiency in wireless networks, highlighting their respective advantages and limitations.
- The preliminary results indicate that the proposed hybrid protocol outperforms the individual algorithms in key metrics, such as data rate and energy consumption, suggesting a more efficient use of resources.
- The findings provide practical guidelines for network operators seeking to implement more effective resource management strategies in real-world wireless communication scenarios.

II. System Model and Problem Formulation

A. System Model

In a single cell, there exists a collection of User Equipment (UE) devices that are tasked with transmitting data. Each device operates under specific power constraints, limiting the amount of transmit power it can utilize. In addition, these devices are allocated a portion of the available bandwidth, which restricts their data transmission capacity.

At the core of this network is a base station responsible for receiving data from all UE devices. This base station not only facilitates communication between the devices but also operates under its own set of power and bandwidth constraints, ensuring efficient data handling and network performance. To enhance this performance, OFDMA is utilized in this network. OFDMA allows multiple users to share the same frequency band by dividing it into numerous orthogonal subcarriers, enabling simultaneous transmission without interference. This approach optimizes bandwidth usage and improves overall system capacity, making it particularly effective in environments with varying channel conditions.

Each UE device has a transmit power used for communication, which is determined by regulatory limits and device capabilities. A finite total bandwidth B_{total} is shared among all devices, necessitating effective scheduling and allocation methods to maximize throughput. OFDMA plays a crucial role in this allocation process, dynamically assigning subcarriers based on user demand and channel quality. Each device can transmit data for a certain time period, contributing to the overall energy consumption. This energy consumption is a critical factor, as it impacts the battery life of the devices and the overall sustainability of the network. Understanding the interplay of these constraints is essential for optimizing performance and ensuring reliable service delivery in wireless communication systems.

1) Data transmission model: The data rate for each device can be modeled using the Shannon capacity formula:

$$R_k = B_k \cdot \log_2 \left(1 + \frac{P_k}{N_0 \cdot B_k}\right), \tag{1}$$

where R_k is the data rate of device D_k, B_k is the bandwidth allocated to device D_k, P_k is the power allocated to device D_k, and N_0 represents the spectral density of the noise power.

2) Energy consumption: The energy consumed by each device during transmission can be calculated as:

$$E_k = P_k \cdot T_k, \tag{2}$$

where E_k is the energy consumed by device D_k and T_k is the transmission time for device D_k.

B. Problem Formulation

The primary objective of the proposed resource allocation model is to maximize the total data rate across all devices while minimizing the overall energy consumption. This can be formulated as a multi-objective optimization problem:

$$\text{Maximize } \sum_{k=1}^{N} R_k - \lambda \sum_{k=1}^{N} E_k, \tag{3}$$

Subject to:

$$\begin{aligned} \text{C1: } & \sum_{k=1}^{N} P_k \leq P_{\text{total}}, \\ \text{C2: } & \sum_{k=1}^{N} B_k \leq B_{\text{total}}, \\ \text{C3: } & \sum_{k=1}^{N} T_k \leq T_{\text{total}}, \end{aligned}$$

where where N represents the total number of devices in the network. R_k represents the data rate for device D_k; P_k denotes the transmission power for device D_k; T_k is the transmission time allocated for device D_k; B_k is the bandwidth allocated to device D_k; and h_k indicates the gain in the channel for device D_k. In constraint C1, the total power allocated to all devices should not exceed the maximum available power. In constraint C2, the total bandwidth allocated to all devices should not exceed the available bandwidth. In constraint C3, the total transmission time allocated to all devices must not exceed the maximum allowed transmission time.

The optimization problem here is mixed integer nonlinear programming (MINLP). Having integer and continuous decision variables leads to this classification. Mixing integers means that some variables, such as the number of UE devices N, can have integer values, while others, such as power P_k, bandwidth B_k, and transmission time T_k, can have continuous values. Additionally, the objective function may include non-linear relationships, such as maximizing total revenue minus a penalty for energy consumption. As a result, if any part of the objective function or constraints shows nonlinearity, the problem is classified as nonlinear. Due to this complexity, specialized algorithms are necessary for finding effective solutions.

III. Optimization Techniques

To effectively solve the optimization problem defined in Eq. (3), population-based algorithms can be employed, including GA, PSO, and hybrid algorithm that combine the strengths of both GA and PSO.

A. Resource Allocation Using GA

GAs utilize principles of evolution to optimize complex problems. They represent candidate solutions as chromosomes, employing a population-based approach where multiple solutions are evaluated concurrently. Each candidate solution is assessed based on an objective function, as defined in Eq. (3).

The selection process in GA strategically prioritizes fitter individuals, allowing them to pass advantageous traits to the next generation through mechanisms such as crossover (where parts of two parent solutions are combined) and mutation (where random alterations are made to a solution). This evolutionary strategy not only improves the quality of solutions, but also fosters genetic diversity within the population, which is essential to avoid local optima. To effectively manage constraints (C1, C2, and C3), GA often implements penalty functions. These penalties reduce the fitness scores of solutions that violate established constraints, thereby discouraging infeasible solutions. This mechanism promotes the exploration of viable solutions while ensuring a diverse search space. Such diversity is crucial for navigating complex non-linear optimization problems, enabling GA to discover high-quality solutions that satisfy all constraints.

A comprehensive overview of the processes involved in GA, including specific steps and methodologies, is shown in Table I.

TABLE I. Optimized Resource Allocation Using GA

Input:

- Population size P: Number of individuals in each generation.
- Number of generations G: Maximum iterations for the GA.
- Crossover rate C_{rate}: Probability of crossover between parents.
- Mutation rate M_{rate}: Probability of mutation for individuals.

Output:

- The best solution representing the optimal resource allocation with its fitness value.

1. **Initialize population:**
Each individual is represented by a random vector:

$$\text{individual}_i = \begin{pmatrix} P_1 & P_2 & \cdots & P_N \\ B_1 & B_2 & \cdots & B_N \\ T_1 & T_2 & \cdots & T_N \end{pmatrix} \tag{4}$$

2. **Evaluate Fitness:**
The fitness of each individual is evaluated based on the objective function:

$$\text{fitness}(\text{individual}_i) = \sum_{k=1}^{N} R_k - \lambda \sum_{k=1}^{N} E_k \tag{5}$$

Ensure that the individual satisfies the constraints C_1, C_2, and C_3. If constraints are violated, apply a penalty to the fitness score:

$$\begin{aligned} &\text{fitness}(\text{individual}_j) = \\ &\begin{cases} \text{fitness}(\text{individual}_j) - P & \text{if constraints violated} \\ \text{fitness}(\text{individual}_j) & \text{otherwise} \end{cases} \end{aligned} \tag{6}$$

3. **Selection:**
   - Select parents based on fitness.
   - Choose a predetermined number of parents to form a mating pool.
4. **Crossover:**
   - For each pair of parents in the mating pool, generate r, which represents a random value generated uniformly within a specific range between $[0, 1]$.
   - If $r < C_{\text{rate}}$, perform crossover to create offspring:

   $$O = \begin{pmatrix} P_1[1:q] \\ P_2[q+1:N] \end{pmatrix} \tag{7}$$

   where q is a randomly chosen crossover point.
5. **Mutation:**
   - Apply mutation to offspring based on the mutation rate:

   $$O[j] = \begin{cases} O[j] + \Delta & \text{if } r < M_{\text{rate}} \\ O[j] & \text{otherwise} \end{cases} \tag{8}$$

   where Δ is a random value drawn from a specified distribution. j-th offspring in the population array O.
6. **Evaluate offspring fitness:**
   - For each offspring, calculate its fitness based on the resource allocation efficiency using equation (5).
7. **Replacement:**
   - Form a new population by selecting the best individuals from both the current population and the new offspring.
8. **Termination:**
   - If the maximum number of generations G is reached or if a satisfactory solution (fitness) is found, stop the algorithm. Otherwise, return to step 3.

B. Resource Allocation Using PSO

Social behaviors are replicated in PSOs by mimicking natural phenomena, such as flocks of birds or schools of fish. In this algorithm, particles represent potential solutions that dynamically adjust their positions in the search space based on both their individual experiences and the collective knowledge of the swarm. Each particle evaluates the objective function to determine its fitness, which guides its movement towards areas of higher quality solutions. The motion of a particle is influenced by two main factors: its previous best position and the best position found by any particle in the swarm. This dual influence allows PSOs to effectively balance exploration (searching new areas of the solution space) and exploitation (known good solutions). Additionally, when particle positions violate established constraints, PSOs typically employ strategies to either adjust the particles back into feasible regions or apply penalties that reduce their fitness scores. This penalty mechanism discourages the swarm from exploring infeasible solutions, thereby maintaining a focus on viable options. PSOs are particularly effective for continuous optimization problems, where the solution space is defined by real valued variables. The algorithm's inherent ability to adaptively explore while converging towards an optimal solution makes it suitable for a wide range of applications, including engineering design, machine learning parameter tuning, and resource allocation. Moreover, the simplicity of PSO, combined with its flexibility, allows it to be easily hybridized with other optimization techniques, further enhancing its performance in complex problem

TABLE II. OPTIMIZED RESOURCE ALLOCATION USING PSO

Input:

- Population size P: Number of particles in the swarm, determining the diversity of solutions.
- Maximum iterations G: The upper limit on the number of iterations for the PSO algorithm, defining its computational duration.
- Cognitive coefficient c_1: Weighting factor that influences how much each particle is attracted to its own best-known position.
- Social coefficient c_2: Weighting factor that influences how much each particle is attracted to the swarm's best-known position.

Output:

- The best solution found by the swarm along with its corresponding fitness value, indicating the effectiveness of the resource allocation.

1. **Initialize Swarm:**
Each particle i is represented with a random position and velocity:

- The position vector for particle i is defined as:

$$\mathbf{X}_i = \begin{pmatrix} P_1 & P_2 & \cdots & P_N \\ B_1 & B_2 & \cdots & B_N \\ T_1 & T_2 & \cdots & T_N \end{pmatrix} \tag{9}$$

where N is the number of devices or resources.

- The velocity vector for particle i is defined as:

$$\mathbf{V}_i = \begin{pmatrix} v_{P1} & v_{P2} & \cdots & v_{PN} \\ v_{B1} & v_{B2} & \cdots & v_{BN} \\ v_{T1} & v_{T2} & \cdots & v_{TN} \end{pmatrix} \tag{10}$$

- Initialize each particle's best position $pbest_i$ to its initial position.

2. **Evaluate Fitness:**

- Calculate the fitness for each particle using the fitness function:

$$\text{fitness}(x_i) = \sum_{k=1}^{N} R_k - \lambda \sum_{k=1}^{N} E_k \tag{11}$$

where R_k represents the reward from resource k and E_k denotes the energy consumption.

- Update the particle's best position $pbest_i$ if the current position yields a better fitness value.
- Ensure that the constraints C_1, C_2, and C_3 are satisfied for each particle's position.

3. **Update global best:**

- Update the swarm's best position $gbest$ based on the best positions found by all particles.

4. **Update velocities and positions:**

- Update the velocity for each particle i:

$$v_i = w \cdot v_i + c_1 \cdot r_1 \cdot (pbest_i - x_i) + c_2 \cdot r_2 \cdot (gbest - x_i) \tag{12}$$

- Update the position for each particle i:

$$x_i = x_i + v_i \tag{13}$$

- Re-evaluate C_1, C_2, and C_3 after updating positions to ensure validity.

5. **Termination:**

- The algorithm stops if the maximum number of iterations G is reached or if a satisfactory solution based on fitness is found.
- If neither condition is met, return to step 2 for further iterations.

domains.

A complete overview of the processes involved in PSO, including specific steps and methodologies, is illustrated in Table II.

C. Resource Allocation using Hybrid Algorithm

In hybrid algorithm, GA and PSO are integrated to improve overall optimization performance. This combination leverages the strengths of both techniques: initial solutions are generated using genetic principles, which emphasize diversity and broad exploration, while subsequent refinements are achieved through swarm intelligence, which focuses on effective local search. By merging these two algorithms, hybrid algorithm can take advantage of the advantages of genetic operations, such as cross-linking and mutation, to explore a wide solution space. This broad exploration is critical for identifying promising regions in complex landscapes. Once potential solutions are identified, the dynamics of the swarm come into play, allowing for precise fine-tuning of these solutions based on the collective knowledge of the swarm. This dual approach improves the convergence speed and the quality of the solution. Moreover, the hybrid algorithm employs penalty mechanisms to handle constraint violations, similar to traditional GA and PSO. By discouraging infeasible solutions, this algorithm maintains a focus on viable options, ensuring that the search remains within the bounds of acceptable solutions. This capability is especially advantageous in complex optimization scenarios where constraints are stringent and multifaceted.

The hybrid approaches are particularly effective for tackling challenging optimization problems, as they can successfully navigate both local and global search spaces. The combination of exploratory genetic principles with the exploitative strengths of swarm intelligence enables this algorithm to escape local optima while still converging toward high-quality global solutions. This versatility makes the hybrid algorithm suitable for a wide range of applications, including engineering design, logistics, financial modeling, and machine learning, where the complexity of the problem demands a robust and adaptive optimization strategy.

A comprehensive overview of the processes involved in the hybrid approach, including specific steps and methodologies, is shown in Table III.

IV. SIMULATION AND RESULTS

The network for the study was simulated using MATLAB, a widely used tool for numerical analysis and data visualization. GA, PSO, and the hybrid algorithm are implemented to assess their effectiveness in resource allocation. The simulation allowed for easy adjustment of parameters and visualization of results in real time, providing insights into network performance under different conditions.

The time complexities associated with solving the resource allocation problem in OFDMA networks vary significantly among the algorithms employed. The GA exhibits a time complexity of $O(P \cdot G \cdot N)$, where P denotes the size of the population, G represents the number of generations, and N corresponds to the number of User Equipment (UE) devices. This complexity arises from the need to evaluate and evolve a population of candidate solutions over multiple generations, each requiring assessment of the fitness of N devices. In parallel, the PSO algorithm demonstrates a time complexity of $O(S \cdot G \cdot N)$, where S indicates the swarm size. As with GA and PSO, the iterative process of updating particle (potential solution) positions based on their own and their peers' experience drives the complexity, requiring evaluations across all N devices in each generation.

In contrast, the hybrid algorithm, which integrates elements of both GA and PSO, incurs a time complexity of $O((P + S) \cdot G \cdot N)$. This reflects the necessity to evaluate the fitness of both populations on each iteration: the particles generated by the GA and the particles from the PSO. As a result, this algorithm may provide a more thorough exploration of the solution space, but at the cost of increased computational complexity. However, all three algorithms exhibit a dependency on the sizes of their respective populations or swarms, the number of generations or iterations, and the number of devices involved. This relationship underscores the computational effort required for larger networks, highlighting a critical consideration for practitioners in the field. As the number of UE devices increases, the time complexity can lead to significant delays in real-time applications. Therefore, optimizing these algorithms or developing hybrid algorithms that can reduce time complexity while maintaining solution quality is essential for scalable and efficient resource allocation in OFDMA networks.

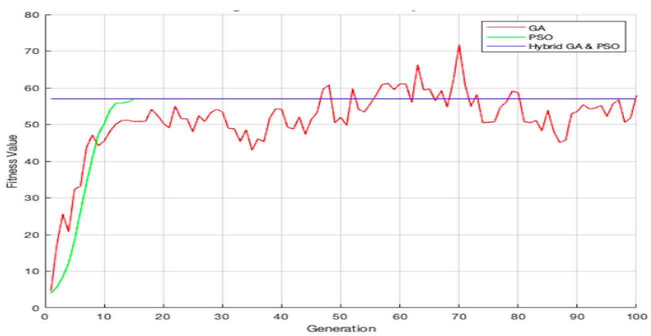

Fig. 1. Comparison of fitness values across generations for GA, PSO, and the hybrid algorithm.

Fig. 1 presents an analysis of the performance of various algorithms in terms of their fitness values. The results clearly indicate that the hybrid approach, which combines GA and PSO, is the most effective in consistently achieving higher fitness values compared to either algorithm used in isolation. The observed performance trends highlight that while PSO tends to provide stable solutions with less variability, the hybrid model effectively leverages the strengths of both algorithms. By utilizing GA's exploratory capabilities to navigate the solution space and PSO's ability to refine and converge on optimal solutions, the hybrid approach demonstrates superior performance across various scenarios. Furthermore, the adaptability of the hybrid algorithm enables it to function effectively in a variety of settings and complex problems, indicating its possible use in real-world situations where environmental changes could occur.

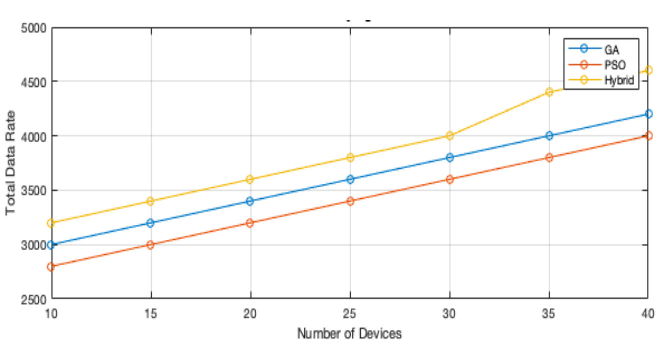

Fig. 2. The total data rate versus the number of devices for GA, PSO, and the hybrid algorithm.

Fig. 2 illustrates a comprehensive analysis of the performance of three algorithms in relation to the total data rate as the number of devices increases. The data reveals a clear trend: all three algorithms demonstrate an improvement in total data rate with the addition of more devices. This improvement highlights the algorithms' ability to effectively utilize available resources as network demand grows. Among the algorithms tested, the hybrid algorithm emerges as the most effective solution, consistently achieving the highest data rates across varying device counts. This superior performance may be attributed to its unique approach, which likely combines the strengths of both traditional and modern techniques, allowing for more adaptable resource allocation and enhanced management of network traffic.

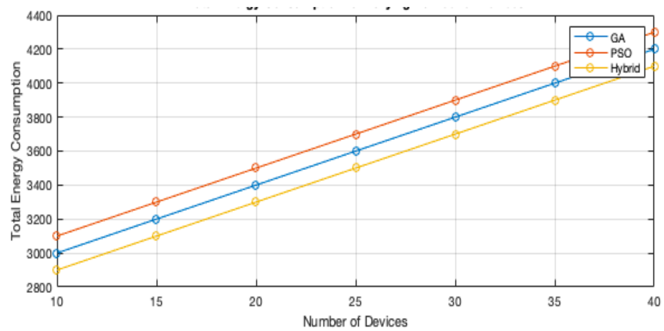

Fig. 3. The power consumption versus the number of devices for GA, PSO, and the hybrid algorithm.

The analysis in Fig. 3 indicates that as the number of devices increases, the energy consumption of all algorithms increases. In spite of its superior data rate performance, the hybrid approach uses the most energy, while GA and PSO exhibit the most efficient energy usage. This shows that trade-offs between energy efficiency and performance, especially in larger networks, need to be carefully considered. In order to better balance these factors, future research could concentrate on improving the hybrid approach. The strong performance of the hybrid algorithm indicates its potential for better resource management and performance optimization, especially in larger networks where device density can significantly affect data transmission efficiency. It could prove a fantastic option for deployment in environments with high user demand because of its capacity to maintain high data rates even as the number of devices increases.

Future work should delve deeper into the specific configurations that contribute to the hybrid algorithm's success. Identifying optimal parameter settings and operational strategies could further enhance its efficacy. Additionally, further investigations into the scalability of this algorithm are essential, as understanding its limits and capabilities in increasingly complex network environments will be crucial for real-world applications. This could involve exploring its performance under varying network conditions, different types of traffic loads, and integration with emerging technologies such as Internet of Things (IoT) and 5G networks.

V. Conclusion

In conclusion, this study highlights the significance of efficient resource allocation in OFDMA networks as a means

TABLE III. OPTIMIZED RESOURCE ALLOCATION USING HYBRID ALGORITHM (GA-PSO)

Input:

- Population size P: Number of individuals/particles in the population.
- Maximum iterations G: Total number of iterations for the hybrid algorithm.
- Crossover rate C_{rate}: Probability of crossover between individuals.
- Mutation rate M_{rate}: Probability of mutation for individuals.
- Cognitive coefficient c_1: Weight for the individual particle's best position.
- Social coefficient c_2: Weight for the swarm's best position.

Output:

- The best solution found by the hybrid algorithm, along with its fitness value.

1. **Initialize population and swarm:**
   - Each individual j is represented by a random vector:
   $$\text{individual}_j = \begin{pmatrix} P_1 & P_2 & \cdots & P_N \\ B_1 & B_2 & \cdots & B_N \\ T_1 & T_2 & \cdots & T_N \end{pmatrix} \quad (14)$$
   - Swarm particles i are represented by a random position and velocity vector:
   $$\mathbf{X}_i = \begin{pmatrix} P_1 & P_2 & \cdots & P_N \\ B_1 & B_2 & \cdots & B_N \\ T_1 & T_2 & \cdots & T_N \end{pmatrix} \quad (15)$$
   $$\mathbf{V}_i = \begin{pmatrix} v_{P1} & v_{P2} & \cdots & v_{PN} \\ v_{B1} & v_{B2} & \cdots & v_{BN} \end{pmatrix} \quad (16)$$
   - Initialize each particle's best position $pbest_i$ to its initial position.
2. **Evaluate fitness:**
   - For each particle i and individual j, calculate fitness based on the objective function:
   $$\text{Fitness}(\mathbf{x}_i) = \sum_{k=1}^{N} R_k - \lambda \sum_{k=1}^{N} E_k \quad (17)$$
   where R_k represents the reward and E_k the energy consumption for task k.
   - Ensure that the individual satisfies the constraints C_1, C_2, and C_3: If constraints are violated, apply a penalty to the fitness:
   $$\text{fitness}(\text{individual}_j) = \begin{cases} \text{fitness}(\text{individual}_j) - P & \text{if constraints violated} \\ \text{fitness}(\text{individual}_j) & \text{otherwise} \end{cases} \quad (18)$$
3. **Update global best:**
   - Determine the best position $gbest$ across all particles and individuals to guide future movements.
4. **Update individuals:**
   - Select individuals based on fitness to form a mating pool.
   - For selected individuals, perform crossover based on C_{rate}:
   $$O[g] = \begin{pmatrix} P_1[1:q] \\ P_2[q+1:N] \end{pmatrix} \quad (19)$$
   where q is a randomly chosen crossover point.
   - Apply mutation based on M_{rate} to introduce variability:
   $$O[g] = \begin{cases} O[g] + \Delta & \text{if } r < M_{\text{rate}} \\ O[g] & \text{otherwise} \end{cases} \quad (20)$$
   where Δ is a specified distribution random value.
5. **Update velocities and positions:**
   - Update velocity for each particle i:
   $$v_i = w \cdot v_i + c_1 \cdot r_1 \cdot (pbest_i - x_i) + c_2 \cdot r_2 \cdot (gbest - x_i) \quad (21)$$
   - Update position for each particle i:
   $$x_i = x_i + v_i \quad (22)$$
6. **Replacement:**
   - Form a new population by selecting the best individuals from both the current population and the new offspring $O[g]$.
7. **Termination:**
   - If the maximum number of iterations G is reached or if a satisfactory solution based on fitness is found, stop the algorithm. Otherwise, return to step 2.

to meet the growing demand for seamless connectivity and energy efficiency. By comparing GA, PSO, and the hybrid algorithm, the hybrid approach effectively balances the benefits of both methodologies, resulting in superior performance in optimizing data transmission and energy consumption. The findings emphasize the importance of adapting resource allocation strategies to the dynamic conditions of modern wireless environments. Future research could explore further enhancements to the hybrid algorithm and investigate its scalability across different network configurations.

REFERENCES

[1] S. C. Yang, *OFDMA system analysis and design*. Artech House, 2010.

[2] Y. O. Imam-Fulani, N. Faruk, O. A. Sowande, A. Abdulkarim, E. Alozie, A. D. Usman, K. S. Adewole, A. A. Oloyede, H. Chiroma, and S. Garba, "5G frequency standardization, technologies, channel models, and network deployment: Advances, challenges, and future directions," *Sustainability*, vol. 15, no. 6, p. 5173, 2023.

[3] P. Mukunthan and P. Dananjayan, "Modified PTS combined with interleaving technique for PAPR reduction in MIMO-OFDM system with different subblocks and subcarriers," *IAENG International Journal of Computer Science*, vol. 39, no. 4, pp. 339–348, 2012.

[4] Z. M. Haider, K. K. Mehmood, M. K. Rafique, S. U. Khan, S.-J. Lee, and C.-H. Kim, "Water-filling algorithm based approach for management of responsive residential loads," *Journal of Modern Power Systems and Clean Energy*, vol. 6, no. 1, pp. 118–131, 2018.

[5] M. Haddad, P. Wiecek, O. Habachi, S. M. Perlaza, and S. M. Shah, "Fair Iterative water-filling game for multiple access channels," in *Proceedings of the 25th international ACM conference on modeling analysis and simulation of wireless and mobile systems*, 2022, pp. 125–132.

[6] B. Wang and X. Hao, "Robust waveform design based on bisection and maximum marginal allocation methods with the concept of information entropy," *Mathematical Problems in Engineering*, vol. 2020, no. 1, p. 3529858, 2020.

[7] A. Eiger, K. Sikorski, and F. Stenger, "A bisection method for systems of nonlinear equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 10, no. 4, pp. 367–377, 1984.

[8] W. Zheng, J. Qiao, L. Feng, and P. Fu, "Optimal cooperative virtual multi-input and multi-output network communication by double improved ant colony system and genetic algorithm," *IAENG International Journal of Computer Science*, vol. 45, no. 1, pp. 89–96, 2018.

[9] J. Qian and G. Chen, "Improved Multi-goal Particle Swarm Optimization Algorithm and Multi-output BP Network for Optimal Operation of Power System." *IAENG International Journal of Applied Mathematics*, vol. 52, no. 3, 2022.

[10] O. Kramer and O. Kramer, *Genetic algorithms*. Springer, 2017.

[11] S. N. Sivanandam, S. N. Deepa, S. N. Sivanandam, and S. N. Deepa, *Genetic algorithms*. Springer, 2008.

[12] J. L. Fernández Martínez and E. García Gonzalo, "The generalized PSO: a new door to PSO evolution," *Journal of Artificial Evolution and Applications*, vol. 2008, no. 1, p. 861275, 2008.

[13] M. Jain, V. Saihjpal, N. Singh, and S. B. Singh, "An overview of variants and advancements of PSO algorithm," *Applied Sciences*, vol. 12, no. 17, p. 8392, 2022.

[14] K. Premalatha and A. M. Natarajan, "Hybrid PSO and GA for global maximization," *Int. J. Open Problems Compt. Math*, vol. 2, no. 4, pp. 597–608, 2009.

[15] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp. 292–305, 2016.