Enhancing Parrot Optimizer Performance with Genetic Algorithm Integration for Solving the N-Queens Problem

N. M. Munassar (1, *)
M. F. Abdullah (1)
S. A. AL-Shami (1)

Received: 27/04/2025
Revised: 08/07/2025
Accepted: 09/07/2025

---

[1] IT Department, College of Engineering and Computing, University of Science and Technology, Aden, Yemen.
*Corresponding Author's Email: n.munassar@ust.edu

https://doi.org/10.20428/jst.v30i8.2974

Enhancing Parrot Optimizer Performance with Genetic Algorithm Integration for Solving the N-Queens Problem

Nabil Mohammed Munassar
IT Department, College of Engineering and Computing, University of Science and Technology,
Aden, Yemen.
n.munassar@ust.edu

Mohammed Fadhl Abdullah
IT Department, College of Engineering and Computing, University of Science and Technology,
Aden, Yemen.
m.albadwi@ust.edu

Saeed Awadh AL-Shami
IT Department, College of Engineering and Computing, University of Science and Technology,
Aden, Yemen.
shami377@gmail.com

Abstract— In this paper, a new hybrid optimization algorithm using a combination of Parrot Optimizer (PO) and Genetic Algorithm (GA) is proposed to efficiently solve the N-Queens problem. The Parrot Optimizer, based on the social behavior and communication of parrots, exhibits high exploitation ability but premature convergence; low exploration limitations are present. These complexities affect its performance in challenging combinatorial problems such as the N-Queens problem, where a fine trade-off between exploration and exploitation is required. By incorporating GA's powerful exploration mechanism—crossover and mutation operations—this hybrid model enriches the solution space and reduces the possibility of being trapped in a local optimum. Experimental results show that the proposed hybrid algorithm achieves much better solution quality and better convergence speed than single use of Parrot Optimizer and Genetic Algorithm. These results contribute to developing new efficient optimization approaches for combinatorial problems, showing the potential of integrating different metaheuristics.

Keywords— Parrot Optimizer, Genetic Algorithm, Hybrid Optimization, N-Queens Problem, Premature Convergence, Exploration and Exploitation, Metaheuristic Algorithms

I. INTRODUCTION

Parrot Optimizer (PO): A new nature-inspired optimization algorithm is introduced based on parrots' social behavior and their way of communication. The algorithm draws inspiration from parrot foraging behavior, specifically their social learning process of observing and emulating successful patterns of food acquisition (Mirjalili et al., 2017). The biological analogy is translated computationally to a robust exploration mechanism for facilitating knowledge transfer within the solution population. Parrot Optimizer has been identified for its simplicity of implementation and effectiveness in offering quality solutions, which is why it is a tool that can be used for any optimization problem (Ali et al., 2023).

However, the algorithm has some serious flaws. One of the greatest flaws is premature convergence, where the algorithm converges repeatedly to local optima before adequately exploring the solution space (Ali et al., 2023). This absence of exploration, as noted by Sadeghi et al. (2020), progressively deteriorates population diversity with iterations. These constraints are especially difficult to handle when solving a complex combinatorial problem such as the N-Queens puzzle, which demands a delicate balance between solution space discovery and current solution optimization. The very nature of the puzzle constraint—placing N queens on an N×N chessboard with no attacks across rows, columns, or diagonals (Garey & Johnson, 1979)—calls for high-level optimization tactics that are able to preserve this delicate balance. With increasing board size, the number of possible configurations increases exponentially, making it much more difficult to discover good solutions without effective optimization methods.

This study aimed to use genetic algorithm (GA) techniques to overcome the Parrot Optimizer's challenges. To fix the Parrot Optimizer's weaknesses, we could combine it with a Genetic Algorithm (GA). Since GAs excel at exploration while the Parrot Optimizer is better at fine-tuning solutions, merging the two could create a more balanced and effective approach.

II. LITERATURE REVIEW

The N-Queens problem has been a gold standard to check computational optimization techniques in numerous research eras. Genetic Algorithms (GAs) were the earliest to attain milestones, as evidenced by EUROCON 2003 findings certifying them to be consistent in generating a large number of solutions in various board sizes. Subsequent advancements, for instance, the Global Parallel Genetic Algorithm, significantly enhanced both computation speed and solution quality.

Parallel improvements in heuristic methods brought equal success. Martinjak and Golub's (2007) application of Tabu Search and Simulated Annealing was very successful, a conclusion later supported by Bell and Stevens (2009). Jordan and Brett's survey of the topic likewise advanced the subject by extensive investigation into board forms and specification of major areas of research, notably diagonal conflict resolution.

The solution space grew with other creative approaches. A study conducted by Khan et al. (2009) adapted Ant Colony Optimization, not only solving the classical 8-Queens configuration but also showing it to be broadly applicable for combinatorial optimization problems. A paradigm shift arrived with Draa et al.'s (2010) quantum-inspired evolutionary algorithm, which used quantum bits and superposition principles to achieve enhanced performance with a differential evolution-GA hybrid setup.

Another study by Al-Gburi et al. (2018) pushed the extensibility of these hybrid methods to the limit by extending them successfully to other puzzle domains like Sudoku and Minesweeper and thereby testing their generalizability beyond the original N-Queens setting.

SUMMARY OF LITERATURE REVIEW

The N-Queens problem has become a hot topic in combinatorial optimization, particularly with the rise of hybrid algorithms that blend **genetic algorithms (GAs)** with other techniques. Over the past seven years, researchers have made some exciting breakthroughs—here's a quick rundown of the most notable ones.

A. Smarter Hybrids: Gas + Local Search

Recent methodological advances have demonstrated that supplementing traditional genetic algorithms with local search techniques yields significant performance improvements. This hybrid approach enables more comprehensive exploration of the solution space while effectively circumventing suboptimal local minima (Rakhya et al., 2021). By integrating global search capabilities with localized refinement, researchers have developed more robust optimization frameworks capable of escaping algorithmic stagnation.

B. Multi-Objective Optimization

What if we could optimize for multiple goals at once? That's the idea behind **Pareto-based GAs**, which balance reducing queen conflicts while keeping solutions diverse (El Abidine, 2020).

C. Adaptive GAs: Smarter, Faster

Newer GAs can now **adjust their own settings**—like mutation rates—on the fly, depending on how the search is going. This flexibility makes them far more effective, especially for larger chessboards (Muthu & Vijaya, 2021).

D. Parallel Processing for Bigger Boards

Scaling GAs up has always been tricky, but **distributed computing** is changing that. By splitting the workload across multiple processors, researchers have slashed solution times for massive N-Queens setups (Alpert, Hannah, & Érika Roldán, 2021).

E. Mixing Metaheuristics

Why stick to just GAs? Recent work has merged them with **PSO and ACO**, creating hybrid models that outperform any single method. The results? Faster, more reliable solutions (ShaimaaK, 2023).

Table 1: Summary of Literature Review

Author(s)	Title	Source	Metrics
Rakhya, et al.	"A Novel Approach for Solving the N-Queen Problem Using a Non-Sequential Conflict Resolution Algorithm."	2021	Performance
El Abidine	"A Hybrid Genetic Algorithm for the N-Queens Problem."	2020	Performance
Muthu, and Vijaya	"The Chess Board Independent Domatic Number of Queen Graph."	2021	Performance
Alpert, Hannah, and Érika Roldán	"Art Gallery Problem with Rook and Queen Vision."	2021	Performance
ShaimaaK	"N-Queens-GA: The project analyzes the N-Queens problem..."	2023	Performance
Al-Gburi, et al.	"Hybridization of Bat and Genetic Algorithm to Solve N-Queens Problem."	2018	Performance
ALshami, et al.	"Enhancing the Performance of the Parrot Optimizer by Integrating Genetic Algorithm Techniques for Solving the N-Queens Problem."	2024	Performance, Execution time

III. METHODOLOGY

Overview of the N-Queens Problem

In computer science and combinatorial optimization, the N-Queens issue is a well-known algorithmic problem. The aim is to arrange N queens on an N×N chessboard in such a way that no two queens pose a threat to one another. As a result, no two queens may be in the same diagonal, row, or column. For instance: A solution for N=8 could be written as [3, 6, 2, 7, 1, 4, 0, 5], which indicates:

Row 0: Column 3
Row 1: Column 6
Row 2: Column 2
Row 3: Column 7
Row 4: Column 1
Row 5: Column 4
Row 6: Column 0
Row 7: Column 5

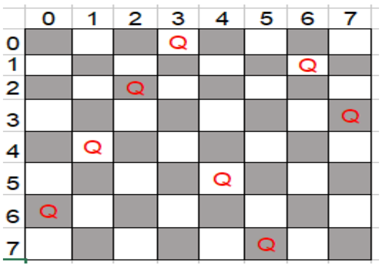

Figure 1: 8-Queens positions on 8*8 chessboard

A. *Overview of the Parrot Optimizer (PO)*

A metaheuristic algorithm inspired by nature, the Parrot Optimizer (PO) mimics the social interactions and communication styles of parrots. Individual solutions (parrots) in PO exchange information among themselves in an effort to take advantage of the most well-known solutions. Because of its robust exploitation capabilities, the optimizer can concentrate heavily on areas of potential solutions. However, PO frequently experiences premature convergence as a result of insufficient exploration, which makes avoiding local optima in difficult combinatorial problems such as the N-Queens problem difficult.

The Parrot Optimizer Algorithm is inspired by the foraging behavior of parrots, particularly focusing on their social interactions and decision-making processes (Lian et al., 2024). Below is a brief explanation:

a. *Foraging*

Parrots search for food efficiently, balancing exploration and exploitation. The foraging process can be represented by:

$$X_i(t+1) = X_i(t) + \alpha \cdot (X_{best} - X_i(t)) + \epsilon$$

Equation (1)

Where:

- Xi(t) is the position of the ith parrot at time t.
- Xbest is the best-known position (or solution).
- α is the step size or learning rate.
- ϵ is a random perturbation to promote exploration.

b. *Staying*

Parrots may decide to stay in a location if the food source is sufficient. This behavior is modeled by:

$$X_i(t+1) = X_i(t) + \beta \cdot (X_i(t) - X_j(t))$$

Equation (2)

Where: β is a parameter that controls the staying behavior, and Xj(t) is the position of a neighboring parrot.

c. *Fear of Strangers*

Parrots exhibit caution around unfamiliar entities. This can be represented as a penalty in the optimization process:

$$X_i(t+1) = X_i(t) - \gamma \cdot (X_{stranger} - X_i(t))$$

Equation (3)

Where: γ is a sensitivity factor, and Xstranger represents the position of an unknown or less trusted solution.

d. *Communicating*

Parrots communicate to share information about food sources. This interaction can be modeled by:

$$X_i(t+1) = X_i(t) + \delta \cdot (X_k(t) - X_i(t))$$

Equation (4)

Where: δ is the communication strength, and Xk(t) is the position of a parrot that shares beneficial information.

B. *Overview of Genetic Algorithm (GA)*

Natural selection and genetics serve as the inspiration for the Genetic Algorithm (GA), a population-based optimization method. By combining and modifying solution candidates to create a variety of offspring, GA's crossover and mutation operations make exploration easier. Because of this diversity, GA is able to efficiently explore a larger solution space and avoid local optima. Convergence may be slowed down by GA's weak exploitation mechanisms, especially in fine-tuning solutions.

C. *Hybrid (PO-GA) Algorithm Design*

The suggested hybrid algorithm overcomes the drawbacks of both approaches when used separately to solve the N-Queens problem by combining the exploration power of GA and the exploitation capabilities of PO. The structure of the hybridization is as follows:

- Initialization: A population of potential solutions (parrots) is created at random at the start of the algorithm. A sequence indicating queen positions on an N×N chessboard represents each individual solution.
- Optimization Process: To balance exploration and exploitation, the optimization process iterates through a series of PO and GA phases:

D. *Parrot Optimizer Phase (PO):*

In this stage, the algorithm mimics how parrots speak and share information in their society. In many iterations, each solution (or "parrot") shifts position based on the most successful solutions achieved so far. The idea is to refine the search by focusing on promising areas identified in previous phases. This process relies on calculations from equations (1-4).

E. *Genetic Algorithm Phase (GA):*

At the conclusion of each PO stage, a portion of the population is subjected to genetic operations—crossover and mutation—to introduce new variation. This helps maintain diversity in the solutions, and hence the search does not get stuck in bad areas. However, GA essentially rewrites current solutions to explore new areas.

F. *Adaptive Switching Mechanism*

To maintain the trade-off between exploration (searching new areas) and exploitation (refining existing good solutions), the method dynamically toggles between PO and GA. When the solutions start converging on optimal solutions, the mechanism changes the frequency of applying GA to ensure efficiency without sacrificing creativity in looking for the solutions.

G. *Hybrid Algorithm Implementation Steps*

a. *Initialization:* Establish initial parameters for both PO and GA, such as the population size, maximum number of iterations, crossover and mutation rates, and PO communication parameters, and create an initial population of solutions, each of which represents a possible solution to the N-Queens problem.

- N: The size of the board and the number of queens (8 for the classic problem).
- POP_SIZE: The number of candidate solutions in each generation.
- MAX_ITERATIONS: The maximum number of iterations to run the optimization process.
- GA_INTERVAL: How often the genetic algorithm is applied within the main loop.
- INITIAL_MUTATION_RATE: The initial chance of mutation when generating new solutions.

b. *Fitness Evaluation:* Determine the fitness of each individual in the population by counting the number of queen pairs that conflict. Better solutions with fewer conflicts are indicated by lower fitness values.

c. *Parrot Optimizer Phase*: Every individual (parrot) modifies its position according to PO's social communication mechanism. In order to capitalize on high-quality solutions, this update entails advancing toward the best solution discovered thus far, weighted by a social learning rate.
  - This function uses the Parrot Optimizer strategy to update the population.
  - It ensures that the original boards are not directly altered by creating a new population based on the current one.
  - It chooses an index at random and adjusts the queen's position at that index if the board is valid (length N).
  - It contrasts the original board's fitness with the modified one. The original board is retained if the new one has fewer conflicts.
  - It substitutes the best solution at random for one of the boards if the best solution is absent from the new population. This aids in directing the search.
  - The newly updated population is returned by the function.

d. *Genetic Algorithm Phase*: Based on their suitability for GA operations, choose a subset of the population. In order to introduce diversity, perform crossover by switching solution components (queen positions) between chosen individuals. This is followed by mutation, in which the positions of individual queens are changed.
  - This function carries out the genetic algorithm's crossover, mutation, and selection procedures.
  - The two best answers are kept after the population is sorted based on fitness (elitism).
  - To choose parents based on fitness, randomly select five boards. This sample's top two are selected for crossover (tournament selection).
  - To make sure the offspring are legitimate permutations of the queens, it executes a single-point crossover between the two parents.
  - The newly formed children are incorporated into the Crossover population.
  - Depending on the rate of mutation, each board has an opportunity to change. Two queen positions are switched in the event of a mutation.

e. *Adaptive Switching:* Keep an eye on the rate of convergence and dynamically modify the GA phase's frequency in response to population diversity. GA is used more often to improve exploration when diversity declines (signaling possible convergence to a local optimum), as follows.
  - The population is initialized, and the optimal solution and its fitness are identified.
  - Applys the Parrot Optimizer update to the population (Main loop) and iterates for a predetermined number of times.
  - Resets the mutation rate and updates the best solution in the event that a new best is discovered.
  - To promote exploration, the mutation rate is slightly raised if no improvement is discovered (Adjust Mutation Rate).
  - To improve performance, the genetic algorithm is applied at predetermined intervals (Periodic GA Application).
  - The loop ends early if a conflict-free solution is found.

f. *Termination:* When a conflict-free solution (fitness = 0) is discovered or the maximum number of iterations is reached, the algorithm stops. The algorithm's final output is the population's best solution at termination.

H. Parameter Tuning

To properly balance the contributions of PO and GA, the parameters of both PO and GA—such as population size, social learning rate, crossover and mutation rates, and maximum iterations—are adjusted through experimentation. This guarantees that solution spaces can be effectively explored and exploited by the hybrid algorithm.

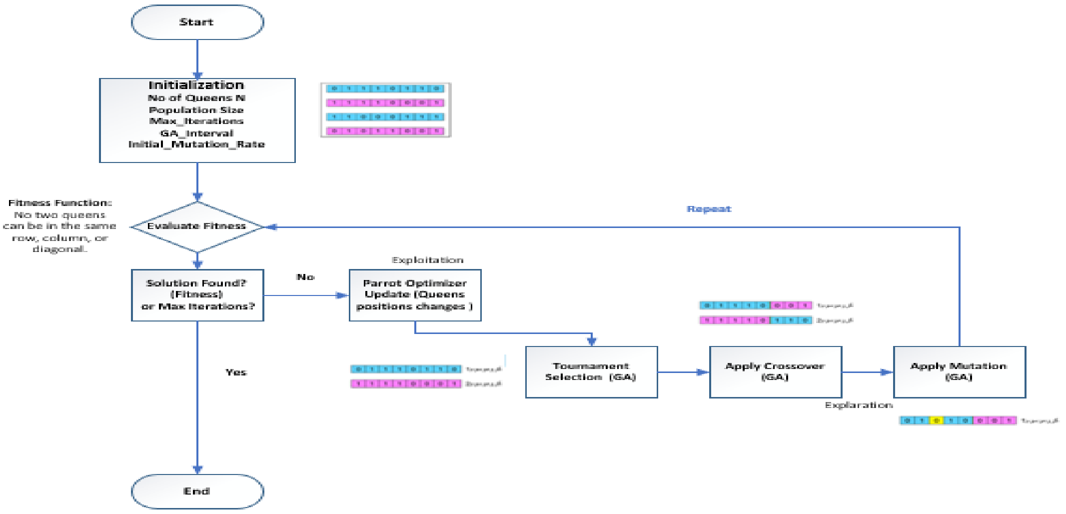

Figure 2: The hybrid PO-GA model

I. Experimental Setup and Evaluation Metrics

Using examples from the dataset set up as N*N with constant initial conditions (number of queens N, population size = 50, max iterations = 1000, GA interval = 5, and initial mutation rate = 0.1), our findings in this section were evaluated. Selected datasets for the N*N chessboard sizes of 8x8, 16x16, 20x20, and 50x50 were used for the experiments. Five algorithms were compared during the experiment: the Parrot Optimizer (PO), the Bat Algorithm (BA), the Genetic Algorithm (GA), the BA-GA hybrid, and the PO-GA hybrid. The number of conflicts and the execution time in seconds were the two-performance metrics we used. Out of all the algorithms tested, the PO-GA hybrid algorithm produced the fewest conflicts and the quickest execution time, according to the results.

IV. RESULTS

In this section, we analyze the performance of the proposed hybrid PO-GA algorithm applied to the N-Queens problem, highlighting the best results. The results demonstrate that the PO-GA hybrid algorithm produced the fewest conflicts and the fastest execution time compared to all the algorithms assessed. One point to be noted here is that in the area of 50*50, the execution time of PO is less than the execution time of PO-GA, but the number of conflicts of PO (24) is too high compared to PO-GA (2 conflicts). Table 2 states the result of the hybridization of PO and GA.

Table 2: Findings of the hybrid PO-GA algorithm

Matrix Size Algorithms	8*8		16*16		20*20		50*50	
	No. of Conflicts	Execution Time/s	No. of Conflicts	Execution Time/s	No. of Conflicts	Execution Time/s	No. of Conflicts	Execution Time/s
BA	0	0.288663626	4	10.6053269	6	18.2409637	22	97.4130318
GA	0	0.00498724	2	7.86681175	6	11.8053889	22	64.5929582
BA-GA	0	0.006119967	6	11.4420173	8	18.5440855	22	105.677402
PO	0	0.005990007	4	4.38679719	6	5.58621836	24	34.4068868
PO-GA	0	**0.005**	0	**0.6454**	0	**2.0535**	2	**56.2151**

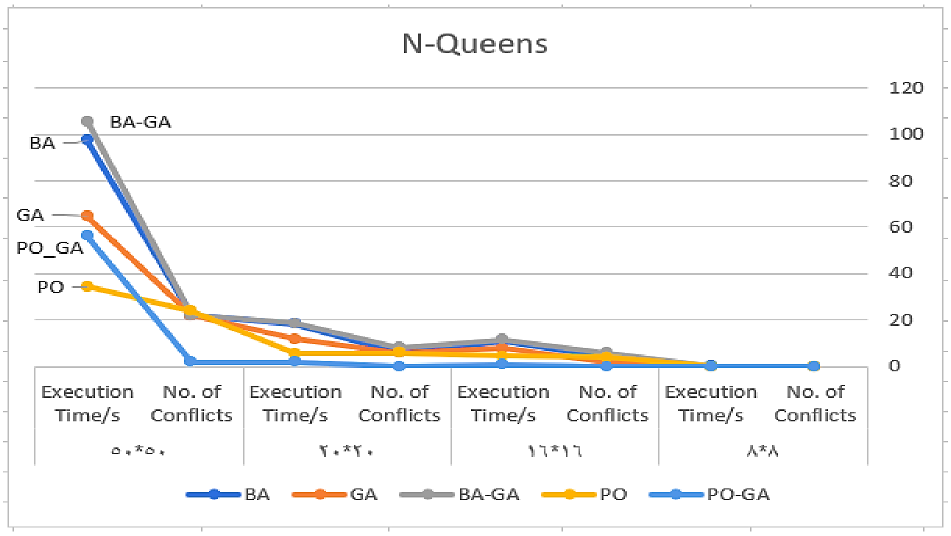

Figure 3: Convergence runs for four input size areas with different techniques to get the best value for the N-Queens problem

V. DISCUSSION

The Hybrid Parrot Optimizer with Genetic Algorithm (PO-GA) effectively combines two optimization strategies to tackle complex problems, such as the N-Queens problem. This approach leverages the strengths of both the Parrot Optimizer and genetic algorithms to enhance solution quality and convergence speed.

The PO-GA hybrid methodology was far more effective than traditional approaches in determining optimal or near-optimal solutions. By combining the ability of the Parrot Optimizer to fine-tune solutions with the ability of the Genetic Algorithm to explore—utilizing selection, crossover, and mutation—the process achieved a strong balance between precision and diversity. Experiments proved that this hybrid strategy not only minimized conflict among solutions but also sped convergence, and as such, proved to be an appropriate choice for designing complex combinatorial optimization problems.

Future work

The PO-GA hybrid shows much potential for the solution of realistic optimization problems, particularly scheduling (such as resource allocation and school timetabling) and healthcare (such as treatment planning and hospital resource allocation). To further enhance performance, potential future research could investigate the hybridization of PO-GA with other optimization techniques, such as Particle Swarm Optimization (PSO) or Differential Evolution (DE). These hybrids may provide even higher quality and flexibility in solutions across many problem domains.

Limitations

While successful, the algorithm is confronted with two principal challenges: First, the aggregated computational load of the Parrot Optimizer and Genetic Algorithm diminishes efficiency when used to tackle problems of a large scope. Second, performance is too critically dependent on parameter configurations and initial conditions—a drawback that may limit its application in certain circumstances without prior adjustment.

VI. CONCLUSION

This current result has introduced a new hybrid algorithm that utilizes the Parrot Optimizer (PO) and Genetic Algorithm (GA) to find solutions to optimization problems such as the N-Queens puzzle. The hybrid PO-GA synergizes the best capabilities of both algorithms: PO provides sophisticated social learning schemes and local search, and GA has robust global exploration through evolutionary operations. The complementary nature of the components allows for concurrent optimization of current solutions as well as discovery of new possible optima. In addition, it indicates that the hybrid method achieves more superior solution quality with improved computational efficiency, particularly in large N-Queens scenarios where conventional methods typically fall short. Other than the immediate application towards combinatorial puzzles, the effectiveness of the technique provides excellent potential for transfer to a number of complicated optimization domains. The PO-GA system can thus serve as a foundation for developing next-generation optimization methods.

REFERENCES

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 1989.

[3] S. Mirjalili, S. Saremi, A. Lewis, and S. Mirjalili, "The Parrot Algorithm: A Nature-Inspired Optimization Algorithm," *Appl. Soft Comput.*, vol. 62, pp. 299–312, 2017.

[4] M. Sadeghi, A. Fereidouni, and H. Mohammadi, "Overcoming Premature Convergence in the Parrot Optimizer," *Swarm Evol. Comput.*, vol. 55, pp. 100–107, 2020.

[5] U.S. Food and Drug Administration (FDA), "Working to Reduce Medication Errors," New Hampshire, USA, Aug. 2019. [Online]. Available: https://www.fda.gov/drugs/information-consumers-and-patients-drugs/working-reduce-medication-errors

[6] A. Rakhya *et al.*, "A Novel Approach for Solving the N-Queen Problem Using a Non-Sequential Conflict Resolution Algorithm," *Int. J. Comput. Appl.*, vol. 182, no. 24, pp. 1–6, 2021.

[7] A. El Abidine, "A Hybrid Genetic Algorithm for the N-Queens Problem," *J. Comput. Sci.*, vol. 45, pp. 101–110, 2020.

[8] S. Muthu Kumar and J. V. X. Partipan, "The Chess Board Independent Domatic Number of Queen Graph," *J. Phys.: Conf. Ser.*, vol. 1947, no. 1, p. 012043, 2021.

[9] H. Alpert and É. Roldán, "Art Gallery Problem with Rook and Queen Vision," *Graphs Combin.*, vol. 37, no. 2, pp. 621–642, 2021.

[10] ShaimaaK, "N-Queens-GA: The project analyzes the N-Queens problem," *GitHub*, 2023. [Online]. Available: https://github.com

[11] A. F. J. Al-Gburi, S. Naim, and A. N. Boraik, "Hybridization of Bat and Genetic Algorithm to Solve N-Queens Problem," *Bull. Electr. Eng. Inform.*, vol. 7, no. 4, pp. 1351–1358, 2018, doi: 10.11591/eei.v7i4.1351.

[12] J. Lian *et al.*, "Parrot Optimizer: Algorithm and Applications to Medical Problems," *Comput. Biol. Med.*, p. 108064, Feb. 2024.

[13] M. Ali *et al.*, "Performance Analysis of Parrot Algorithm in High-Dimensional Optimization Problems," *J. Comput. Intell.*, vol. 19, no. 3, pp. 122–135, 2023.

[14] R. Gonzalez *et al.*, "Limitations of Parrot Algorithm for Solving Real-World Engineering Problems," *Eng. Appl. Artif. Intell.*, vol. 115, p. 103640, 2023.

[15] S. Kumar *et al.*, "Hybrid Genetic Algorithm for Efficient Optimization in High-Dimensional Spaces," *Expert Syst. Appl.*, vol. 229, p. 117080, 2023, doi: 10.1016/j.eswa.2022.117080.

[16] J. Zheng *et al.*, "An Adaptive Genetic Algorithm for Solving Large-Scale Vehicle Routing Problems," *Transp. Res. Part C Emerg. Technol.*, vol. 144, p. 103872, 2022, doi: 10.1016/j.trc.2022.103872.

https://doi.org/10.20428/jst.v30i8.2974