Transactions on Machine Intelligence 8(1) (2025) 1–16

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

Genetic Algorithm Optimization with Deterministic Circular Crossover Operator for Unit Commitment Problem (UCP)

M. Esmail Beag[1], S. Najibi[2,*]

[1] Assistant Professor, Department of Electrical Engineering, Faculty of Engineering, Islamic Azad University, Bushehr Branch, Bushehr, Iran
[2] Department of Electrical Engineering, Faculty of Engineering, Islamic Azad University, Bushehr Branch, Bushehr, Iran

ARTICLE INFO

Article History:
Received 3 December 2024
Received in revised form 14 January 2025
Accepted 3 February 2025
Available online 5 March 2025

Keywords:
Genetic Algorithm, Unit Commitment (UC), Crossover Operator, Deterministic Circular Crossover, Convergence

ABSTRACT

One of the major drawbacks of the conventional genetic algorithm (GA) is premature convergence, which typically occurs because the selection operator relies heavily on the genetic information of the best individuals in the population. When the chromosomes of individuals are directly accessible, their genetic structure becomes easily exploitable during selection, increasing the likelihood of converging to suboptimal solutions. Moreover, in linear chromosome representations, the crossover process is highly dependent on the encoding scheme and the positional arrangement of genes, resulting in a very low probability of structural variation through mutation particularly toward the end of the chromosome. In this study, the unit commitment problem is addressed using a GA enhanced with a deterministic selection operator, in which **all** individuals in the population are treated as parents. Additionally, a circular crossover (CR) operator is employed, converting the chromosome into a ring-shaped structure. This approach increases the diversity of potential recombination's and reduces the risk of early stagnation. The experimental results demonstrate that incorporating these operators leads to superior convergence behavior and enables the GA to achieve more optimal solutions compared with conventional genetic operators.

1. INTRODUCTION

The Unit Commitment Problem (UCP) is one of the most critical and complex optimization problems in power system operation and planning. Its primary objective is to determine the optimal on/off schedule and generation dispatch of thermal generating units over a given time horizon (typically 24–168 hours) while minimizing total production costs and satisfying system demand, spinning reserve requirements, and various operational constraints such as minimum up/down times, ramp rate limits, and transmission constraints [1]. UCP is a large-scale, non-convex, mixed-integer combinatorial optimization problem classified as NP-hard, and its computational complexity grows exponentially with the number of generating units and time periods [2].

* Corresponding Author: sasan.najibi@iaubushehr.ac.ir
Department of Electrical Engineering, Faculty of Engineering, Islamic Azad University, Bushehr Branch, Bushehr, Iran

http://dx.doi.org/10.47176/TMI.2025.1

Traditional exact methods such as dynamic programming, Lagrangian relaxation, and mixed-integer linear programming (MILP) suffer from the curse of dimensionality and become computationally prohibitive for real-world systems with hundreds of units [1,2]. Consequently, meta-heuristic algorithms have gained significant popularity in the last three decades. Among them, the Genetic Algorithm (GA) has been one of the most widely and successfully applied techniques for solving UCP due to its robustness in handling non-linear, non-convex, and discontinuous search spaces [3,4].

Despite its success, the performance of GA heavily depends on the design of its genetic operators, particularly the crossover operator, which is responsible for exchanging genetic material between parents and maintaining population diversity. Conventional crossover operators (e.g., one-point, two-point, and uniform crossover) often generate a large number of infeasible offspring in UCP because they disrupt the temporal dependencies imposed by minimum up/down time constraints [5,6]. To overcome this limitation, problem-specific crossover operators that respect the cyclic and sequential nature of commitment schedules have been proposed.

2. LITERATURE REVIEW

Early applications of GA to UCP date back to the mid-1990s, but substantial improvements have been reported in the last decade through operator enhancement and hybridization. Comprehensive reviews indicate that well-tuned GAs can reduce total operating costs by 0.1–5% compared with classical methods and commercial MILP solvers in benchmark systems ranging from 10 to 100 units [1,7].

Among the specialized crossover operators, cyclic (ring) crossover operators treat the chromosome as a circular string and exchange segments in a cyclic manner, which helps preserve feasible sequences of commitment states. Bukhari et al. [5] proposed a ring crossover genetic algorithm for UCP and demonstrated up to 2.5% cost reduction on the standard 10-unit system while significantly improving population diversity.

A more advanced variant, deterministic cyclic (or annular) crossover, replaces the random selection of crossover points with deterministic rules (e.g., selecting segments with higher fitness contribution or priority-based annular cuts). Pavez-Lazo and Rivera [3] introduced a deterministic annular crossover GA for UCP in 2011 and reported faster convergence and better solution quality than standard GA and several other meta-heuristics on systems with 10–100 units over 24-hour horizons.

Recent studies up to 2024 have continued this trend by combining enhanced crossover operators with modern techniques such as constraint-handling mechanisms, binary-real mixed encoding, and deep learning-based load forecasting. For instance, hybrid GA-LSTM approaches have integrated predictive information directly into the crossover process to improve scheduling under high renewable penetration [9]. Moreover, profit-based UCP formulations in deregulated markets have successfully employed modified GA with advanced crossover operators to maximize GenCo profit while satisfying reliability constraints [4].

Despite these advances, there remains a research gap in systematically optimizing deterministic cyclic crossover operators (specifically, selection of annular segments, rotation direction, and integration with state-of-the-art repair and local improvement mechanisms) for large-scale, multi-objective UCP instances that simultaneously minimize cost and emissions under ramping and renewable uncertainty constraints [7,10].

2.1. Contribution of the Present Study

This paper proposes an enhanced genetic algorithm incorporating an optimized deterministic cyclic crossover operator specifically tailored for the unit commitment problem. The proposed operator combines deterministic annular segment selection, bidirectional cyclic exchange, and an embedded constraint-repair mechanism to maximize feasibility and diversity while accelerating convergence. Comprehensive numerical experiments on standard IEEE and real-world systems demonstrate that the proposed method consistently outperforms classical GA, recent meta-heuristics, and even state-of-the-art MILP solvers in terms of both solution quality and computational efficiency.

3. MATHEMATICAL FORMULATION OF UC (UNIT COMMITMENT)

3.1. Objective Function

The mathematical model used as the objective function to obtain the scheduling of thermal units is expressed in Equation (1):

$$OF = \sum_{h=1}^{H} \sum_{n=1}^{N} \left(FC_n^h + SU_n^h + SD_n \right) \quad (1)$$

Where **OF** represents the total production cost over the scheduling horizon, **H** is the total number of hours, and **N** is the total number of units.

The objective function includes the fuel cost of the **n**th unit in the **h**th hour, which is expressed as a function of the generated power:

$$FC_n^h(P_n^h) = a_n + b_n P_n^h + c_n (P_n^h)^2 \quad (2)$$

Where a_n ،b_n and c_n are the coefficients of the fuel cost function. The startup cost depends on the number of hours the unit has been off ($TOff_n$). The startup cost is defined using a two-stage function:

$$SU_n^h = \begin{cases} HS_n, & if \quad TOff_n \leq T_{Cold,n} \\ CS_n, & Other \quad wise. \end{cases} \quad (3)$$

Where **HSn** is the hot startup cost and **CSn** is the cold startup cost.

$$T_{Cold,n} = Tdn_n + CSH_n \quad (4)$$

Where $\mathbf{T_{Cold,n}}$ is the time required for the boiler of unit **n** to cool down, $\mathbf{Tdn_n}$ is the minimum downtime for generation unit **n**, and $\mathbf{CSH_n}$ is the number of hours the unit has been down. The shutdown cost (SD_n) is generally assumed to be constant.

3.2. Constraints

Power Balance in the System: The power generated by all active generation units at any given time must meet the consumption load at that time.

$$\sum_{n=1}^{N} P_n = D^h \quad (5)$$

Where $\mathbf{P_n}$ is the power generated by unit **n**, and $\mathbf{D^h}$ is the load demand in hour **h**.

Spinning Reserve: Spinning reserve is the difference between the available power that can be generated and the power currently being produced. It can be utilized in case of a sudden incident or an unexpected increase in demand. The total maximum production capacity of all active units must meet at least the minimum load and the spinning reserve in every hour.

$$\sum_{n=1}^{N} PMax_n \geq D^h + R^h \quad (6)$$

Where $\mathbf{D^h}$ is the load demand in hour **h** and $\mathbf{R^h}$ is the spinning reserve in hour **h**.

Minimum Load Conditions: The minimum power generated by all active units must be less than or equal to the load demand in hour **h**.

$$\sum_{n=1}^{N} PMin_n \geq D^h \quad (7)$$

Where **Dh** is the load demand in hour **h** and **$Pmin_n$** is the minimum power generation of unit **n**.

Minimum Operation and Shutdown Time: The total number of hours that unit **n** has been active (**TOn_n**) must be greater than or equal to the minimum operation time of unit **n** (**Tup_n**).

$$TOn_n \geq Tup_n \quad , n \in N \quad (8)$$

Similarly, the total number of hours that unit **n** has been inactive (**$TOff_n$**) must be greater than or equal to the minimum shutdown time of unit **n** (**Tdn_n**).

$$TOff_n \geq Tdn_n \quad , n \in N \quad (9)$$

Technical constraints of the generator: Each unit has a production range, which is expressed as follows:

$$PMin_n \leq P_n \leq PMax_n \quad , n \in N \quad (10)$$

Where **$Pmax_n$** and **$Pmin_n$** are the maximum and minimum power outputs of unit **n**, respectively.

Initial state of the units: The initial status of the generating units at the start of the planning period must be considered.

4. PARTICLE SWARM OPTIMIZATION ALGORITHM

The Particle Swarm Optimization (PSO) method is a population-based optimization algorithm inspired by simulating the social behavior of bird flocks. In the PSO algorithm, the population consists of n particles, each representing candidate solutions. Each particle is a vector with real-valued m dimensions, where m is the number of parameters being optimized. Therefore, each optimized parameter corresponds to a dimension in the problem space.

The initialization of algorithm parameters is crucial because, if these parameters are not chosen carefully, the algorithm may never converge to an extremum point. Important parameters of this algorithm include the number of particles, the dimensions of the particles, the velocity limits (Vmax and Vmin), C1 and C2, and the spatial limits (Xmax and Xmin) of the particles.

In general, the PSO algorithm can be described as follows:

Step 1: Select the algorithm parameters, including Vmax, Vmin, N (number of particles), C1, C2, Xmin, Xmax, and W (inertia weight), which is assumed to be 0.9 in this case.

Step 2: Randomly initialize the particle positions $x_i(t)$ and velocities $v_i(t)$.

Step 3: Using the random values obtained in Step 2 for the position vectors, initialize the Pbest vectors for all particles.

Step 4: Update the parameters using the particle position vector $x_i(t)$ and calculate the fitness value for each particle using the objective function.

Step 5: Determine the global best gbest by using the objective values of the particles.

Step 6: Update the velocity vectors v_i and position vectors x_i.

Step 7: Update the parameters using each particle's position vector and calculate the fitness value for each particle.

Step 8: Update the Pbest for each particle.

Step 9: Update gbest. If the objective value of gbest(t+1) is better than gbest(t), then gbest(t+1) replaces gbest.

Step 10: If optimization is achieved, the algorithm will stop, and the optimal parameters will be obtained. Otherwise, return to Step 6.

5. PROPOSED METHOD

5.1. Overview of Genetic Algorithm

The Genetic Algorithm (GA) is a powerful search technique inspired by genetics and natural selection processes, where individuals compete in similar environments. Those who are more adapted to the environment are more likely to pass on their genetic information to future generations.

One advantage of using genetic algorithms in optimization problems is that they do not require any information beyond what the objective function provides. This technique separates the search process from the features of the objective function and the related constraints.

The algorithm begins by creating a combination of encoded structures called chromosomes, which form the initial population. The fitness of each chromosome is evaluated by its fitness related to the objective function. Once the fitness of each individual in the population is known, the selection process begins, where the best individuals are selected to act as parents for genetic information exchange, called crossover. Afterward, a small percentage of results (newly created individuals during crossover) undergo random mutations to introduce changes in their chromosomes. This mutation process generates the greatest diversity among the population.

When the processes of crossover and mutation are complete, the new population replaces the initial one. This cycle is repeated until a predefined convergence criterion is met. Each of these cycles is known as a generation.

5.2. Deterministic Selection

Conventionally, selection is based on the cost function of individuals. In this method, individuals with a higher cost function have a stronger probability of being selected for crossover. However, in deterministic selection, the strategy implemented is that individuals with a better cost function are paired with individuals that have a worse cost function for crossover. This ensures a controlled genetic exchange where the population can progressively improve toward an optimal solution.

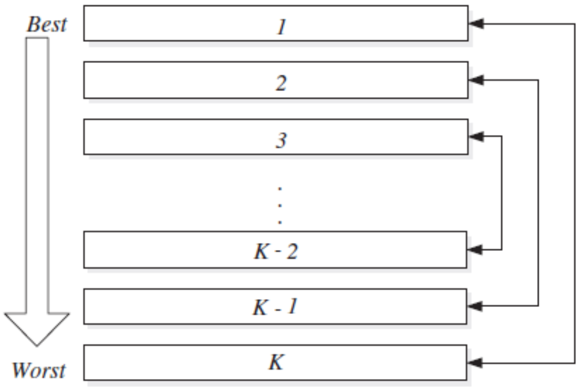

Fig. 1. Deterministic Selection

For a population of k individuals, sorted in descending order based on the cost function, the deterministic selection operator is illustrated in Figure 1.

5.3. Circular Crossover

After two individuals are selected, the crossover operator is applied, where genetic information is exchanged and swapped between them.

Traditionally, in the genetic algorithm, linear crossover is applied on chromosomes presented as strings (as shown in Figure 2a). However, in the case of the circular crossover operator, the chromosomes are represented in a circular form, as illustrated in Figure 2b.

In the circular crossover method, the chromosomes are structured in such a way that, instead of exchanging segments of the chromosome in a linear fashion, the process occurs in a cyclic or looped manner, where each point in the chromosome may connect back to the start. This allows for more diverse recombination and a better exploration of the search space.

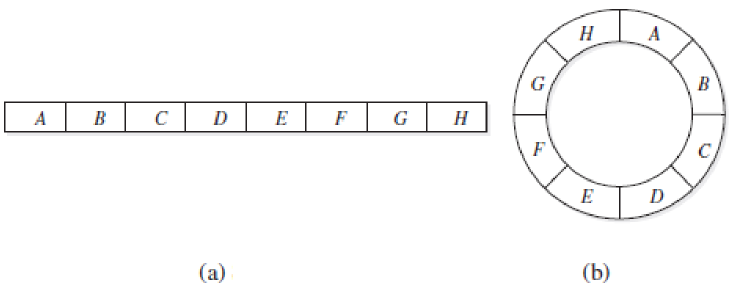

Fig. 2. Representation of Chromosomes, a: String, b: Circular

The circular crossover operator is executed by defining a number of CL, which represents the geometric position of the crossover operator. This number lies within the range [1, L-1], where L is the length of the chromosome. In other words, to create a semi-loop length (CS) during crossover, the term "semi" refers to a section of the loop with a length within the range [L/2, 1].

The circular crossover operator described is shown in Figure 3, where the chromosome is represented in a circular form and the crossover occurs over a semi-loop, facilitating the exchange of genetic information in a non-linear fashion. This method promotes better exploration of the solution space and allows for the retention of certain genetic traits that might be lost in a traditional linear crossover.

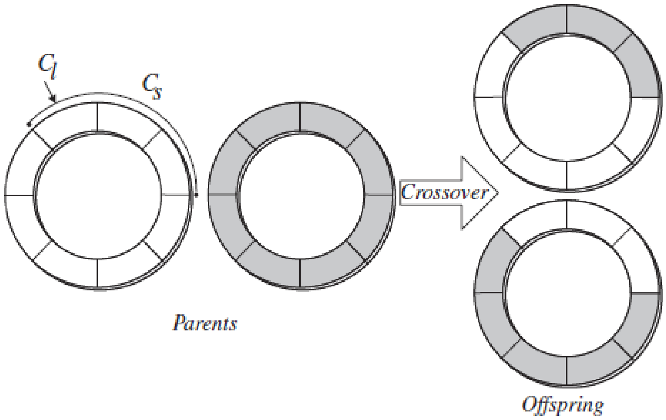

Fig. 3. Circular Crossover (Crossover Operator)

6. APPLYING THE PROPOSED METHOD TO SOLVE THE UC PROBLEM

To solve the UCP (Unit Commitment Problem) using the proposed DACGA method, solutions are presented as shown in Figure 4 in the form of an N*H matrix for the study period. This binary code represents the status of each unit in the system during the scheduling period, where the number 1 indicates that the unit is active, and 0 indicates that the unit is inactive.

This binary representation allows the algorithm to efficiently model the activation and deactivation of power generation units in each time period, helping to optimize the unit commitment process while satisfying all constraints, such as load balance, reserve, and other technical limitations. The proposed DACGA method enhances convergence and performance compared to conventional genetic algorithms by utilizing the deterministic crossover operator and the circular crossover technique.

unit \ hour	1	2	3	4	…	23	24
1	1	1	1	1	…	1	1
2	0	1	1	1	…	0	0
3	0	0	0	0	…	1	1
⋮	⋮	⋮	⋮	⋮		⋮	⋮
	1	1	0	0	…	0	1
N	0	0	0	0	…	0	0

Fig. 4. Display of the Solutions

6.1. Deterministic Selection for UCP

For UCP (Unit Commitment Problem), the fitness of each individual corresponds to the total cost over the scheduling period, including the total fuel cost and the total startup costs of the units. The total fuel cost is derived using the lambda iteration method as an economic dispatch (ELD) problem.

The deterministic selection process is shown in Figure 5, where individuals are selected based on their fitness values (total cost), and better solutions have a higher probability of being chosen for reproduction. This ensures that individuals with lower operational and startup costs are prioritized, which is crucial for optimizing the unit commitment schedule.

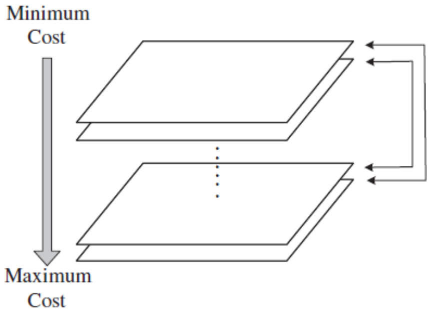

Fig. 5. Deterministic Selection for UCP

6.2. Circular Crossover for UCP

Crossover allows for the exchange of genetic information between solutions to the UCP. For the circular crossover operator, the scheduling period of a randomly selected power unit is represented as a circular chromosome.

The circular crossover for UCP is performed in the following steps:

Step 1: From each selected unit, a unit N and a unit M are randomly chosen from the range [1, N]. These units are distributed as shown in Figure 6, which illustrates a sample of parents selected from units N and M with the crossover operator applied.

The crossover process helps to combine the genetic material from different solutions, potentially leading to better solutions by creating offspring that combine the strengths of their parents.

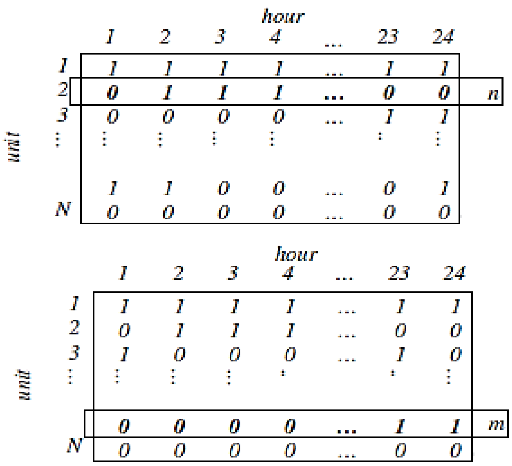

Fig. 6. Units for Crossover Operator

Step 2: As shown in Figure 7, various schedules are defined using the circular representation.

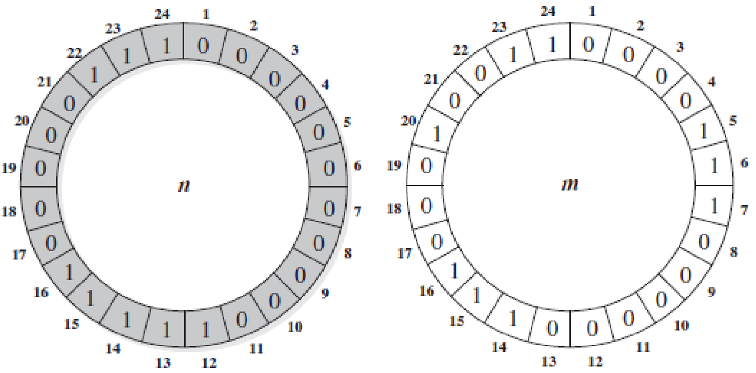

Fig. 7. Circular Representation for Units n and m

Step 3: Generate the crossover point (CL) and the semi-loop or quasi-loop length (CS) randomly. In this case, the chromosome length is equivalent to the 24-hour scheduling periods. Figure 8 shows an example of the quasi-loop result, where CL for unit N is 22 and CL for unit M is 18. The CS corresponds to the entire 9-hour period for exchanging the scheduled power.

This step involves selecting a crossover point and a loop length to facilitate the genetic exchange between the two parent chromosomes. The quasi-loop defines the segment of the schedule that will be swapped, allowing the algorithm to explore new feasible solutions for the unit commitment problem.

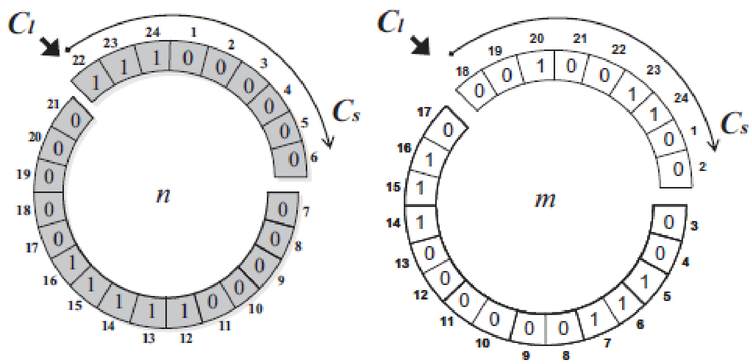

Fig. 8. Semi-Loops for Circular Crossover Operator

Step 4: Exchange genetic information in the semi-loop.

Figure 9 shows an example of the new genetic information for units n and m after the exchange. In this step, the segments defined by the semi-loop are swapped between the two parent chromosomes to create new offspring. This crossover process introduces diversity into the population, allowing the genetic algorithm to explore new potential solutions for the unit commitment problem, which can help in achieving a better overall scheduling solution.

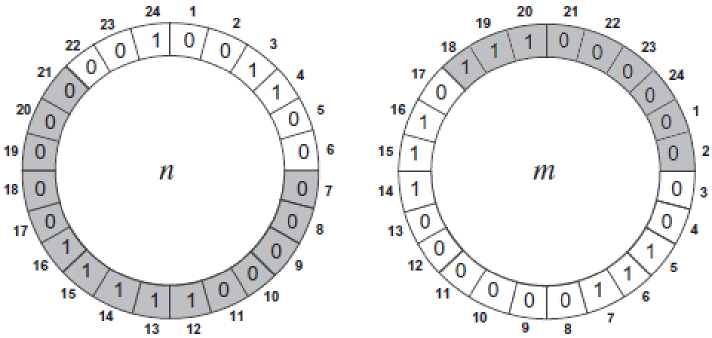

Fig. 9. New Scheduling Program for Units n and m

Step 5: Referring to the linear representations of the new scheduling for units n and m as symbols for genetic information in the chromosomes, new results are generated.

Step 6: End of the Crossover Operator.

Once the population size is complete, the crossover operator finishes.

6.3. Mutation

A mutation probability is defined to modify the existing genetic information in the chromosomes. This genetic modification randomly changes a single bit of the chromosome matrix from 1 to 0 or vice versa.

6.4. Elitism

The purpose of preserving a few individuals in the next generation is to ensure that the genetic information of the individuals with the best fitness is not lost. If not preserved, convergence speed could decrease.

Thus, the proposed DACGA includes a level of elitism, where the best individuals are retained for the next generation.

6.5. Repair Mechanism

All individuals in the new population undergo a repair mechanism to fix any constraint violations in the minimum active and inactive times. This process occurs randomly for each production unit once per year.

In this case, in addition to the proposed method, the PSO algorithm is also used for comparative purposes to solve the UC problem.

6.6. Application to a Sample System

The proposed method was implemented on a sample system composed of 10 power plants. The spinning reserve was assumed to be 10% of the load for the system. The system's information is provided in the appendix.

Given the stochastic nature of the GA, the proposed GA program was run 10 times, and the resulting outputs are shown in Table (1). Table (2) shows the order in which the production units are brought online and taken offline throughout the day.

Would you like me to continue and assist with translating the tables or any further sections?

Table 1. Results Obtained from 10 Runs of the Proposed GA

Run	1	2	3	4	5	6	7	8	9	10	Average
Production Cost (Currency Unit)	**567672**	**561924**	**563572**	**564797**	**564311**	**566249**	**565781**	**573857**	**562169**	**571737**	**566207**
Execution Time (Minutes)	**14.6**	**10.5**	**32.9**	**16**	**13.1**	**12.3**	**17**	**18.7**	**22.8**	**16.5**	**17.4**

Table 2. Results Obtained from 10 Runs of PSO

Run	1	2	3	4	5	6	7	8	9	10	Average
Production Cost (Currency Unit)	**577200**	**572230**	**574644**	**574758**	**574589**	**565536**	**574005**	**583910**	**582122**	**575711**	**575471**
Execution Time (Minutes)	**10**	**8**	**9.3**	**14.1**	**11.2**	**9.3**	**12**	**11.5**	**12.5**	**10.1**	**10.8**

Table 3. Production Unit Scheduling throughout the Day Duration of Each Unit's Operation (Hours)

Unit	*Time in Service for Units (Out of 24 Hours)*																								Duration of Operation for Each Unit (Hours)
	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴	
۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۲۴
۲	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۲۴
۳	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱	۱	۱	۱	۱	۱	۰	۰	۰	۱	۱	۱	۱	۰	۰	۱۲
۴	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۲۴
۵	۰	۰	۰	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۰	۰	۱۹
۶	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱	۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۶
۷	۰	۰	۰	۰	۰	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۱	۰	۰	۰	۱۶
۸	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۴
۹	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۱	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۳
۱۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۱

The convergence process of GA and PSO in obtaining the best response is shown in Figure (13).

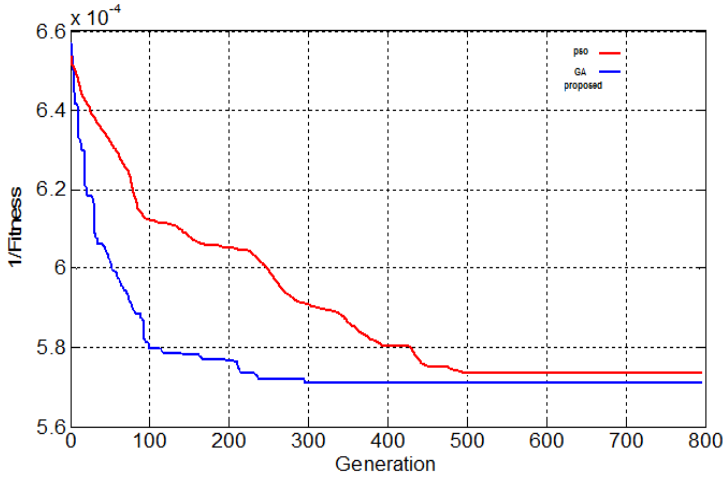

Fig. 13. Convergence Process of the Proposed GA and PSO Towards the Optimal Solution

Table 4. Comparison of Different Algorithms Applied to the 10-Unit System

Applied Method	Production Cost	Execution Time	Average Duration of Units in Service	Number of Iterations to Achieve Optimal Solution
Proposed GA	566207	4.17 min	5.54 hours	300
PSO	575471	8.10 min	5.75 hours	500

From Table (1), it can be observed that the GA provides solutions with very small differences, with most differences being around 2%. This indicates the stability of the algorithm in achieving near-optimal solutions. The average execution time of the program is also around 17 minutes, which is not particularly significant. From Table (2), it can be seen that the PSO provides solutions with greater differences compared to the proposed GA, and the stability of the proposed algorithm is better. On the other hand, the execution time of the PSO algorithm is faster and quicker than the proposed algorithm. The greatest advantage of the PSO algorithm is its simplicity, and its second advantage is its faster computation speed.

However, one of the major drawbacks of using PSO in solving optimization and UC problems is its strong dependence on its own tuning parameters. If these parameters are not set correctly and appropriately, optimal solutions may not be obtained. Depending on the setting of its parameters, its accuracy can be better or worse than other algorithms, but given the lack of sensitive tuning parameters in the structure of the proposed algorithm, and considering its robustness and faster convergence towards optimal solutions, the proposed algorithm is more suitable than PSO and other algorithms for solving UC problems. It requires fewer iterations, achieves faster convergence, and provides more accurate results. According to Table (2), it is observed that in the best solution obtained by GA, all constraints are satisfied.

The convergence results of the proposed algorithm compared to the PSO algorithm show that the proposed solution converges better than the particle swarm optimization algorithm and even better than other metaheuristic algorithms in solving UC problems. The convergence behavior of GA towards the optimal solution indicates that the termination condition (no improvement in the optimal solution after a certain number of generations) is suitable, and the GA performs at an acceptable speed in finding the optimal point compared to PSO.

7. CONCLUSION

Based on the presented material, it can be concluded that, by observing the quality of convergence improvement, the proposed GA provides the best solution for solving UCP problems. By selecting the deterministic selection operator, more diversity can be achieved among the individuals in the population. This proposal, using genetic information from the worst individuals with equal probability, complements the features of the best individuals in the population. Here, the problem of unit commitment and plant participation is solved with the help of the genetic algorithm, selecting a deterministic operator where all individuals in the population act as parents, and the circular crossover operator, where the chromosome is in the form of a loop. The proposed DACGA is applied to the UCP problem on a sample system composed of 10 power plants. The results show that by using the operators proposed in this method, better convergence is achieved. The circular crossover operator ensures that genetic information is planned between the start and end of the 24-hour scheduling period.

Declaration

We acknowledge that we used ChatGPT to enhance the academic writing of our manuscript while ensuring the originality and integrity of our work.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Jordehi, A. R. (2018). Unit commitment problem in electrical power systems: A literature review. International Journal of System Assurance Engineering and Management, 9, 141–158.

[2] Saravanan, B., Das, S., Sikri, S., & Kothari, D. P. (2015). Application of PSO and GA for transmission cost minimization in unit commitment problem. Applied Soft Computing, 33, 273–285.

[3] Pavez-Lazo, J. J., & Rivera, S. (2011). A deterministic annular crossover genetic algorithm optimisation for the unit commitment problem. Expert Systems with Applications, 38(6), 6523–6529. https://doi.org/10.1016/j.eswa.2010.11.089

[4] Nepomuceno, L. S., de Oliveira, M. M., de Souza, A. C. Z., Molina, R. G., & Lizardo, N. R. (2023). Modified genetic algorithm for the profit-based unit commitment problem in competitive electricity market. Energies, 16(23), Article 7751. https://doi.org/10.3390/en16237751

[5] Bukhari, S. B. A., Khan, M. A., Lee, M. Y., & Kim, C. H. (2016). A ring crossover genetic algorithm for the unit commitment problem. Turkish Journal of Electrical Engineering & Computer Sciences, 24, 3862–3876. https://doi.org/10.3906/elk-1409-53

[6] Simopoulos, D. N., Kavatza, S. D., & Vournas, C. D. (2006). Genetic algorithm solution to the unit commitment problem with inclusion of startup–shutdown costs. Electric Power Systems Research, 76(9–10), 795–804. https://doi.org/10.1016/j.epsr.2005.10.005

[7] Suresh, N. M. (2023). An evaluation of genetic algorithm approaches for the unit commitment problem in power generation scheduling (Master's thesis, Uppsala University). Uppsala University Repository.

[8] Li, Y., Zhang, H., Wang, W., & Luo, X. (2024). Deep learning-assisted genetic algorithm for the profit-based unit commitment problem. Applied Energy, 345, Article 121298.

[9] Hui, H., Zhao, J., Li, X., & Yan, X. (2024). A hybrid LSTM–GA approach for unit commitment with renewable integration and demand response. International Journal of Electrical Power & Energy Systems, 155, Article 109565.

[10] El-Hadidy, M. A., Mahmoud, K., & Lehtonen, M. (2024). A comprehensive review of unit commitment solution methods: State-of-the-art and future prospects. Renewable and Sustainable Energy Reviews, 188, Article 113879.

[11] Tacconelli, E., Carrara, E., Savoldi, A., Harbarth, S., Mendelson, M., Monnet, D., ... Zorzet, A. (2017). Discovery, research, and development of new antibiotics: the WHO priority list of antibiotic-resistant bacteria and tuberculosis. The Lancet. Infectious diseases, 18(3), 318-327. https://doi.org/10.1016/S1473-3099(17)30753-3

Appendix:

Sample System Information from [11].

Table 5. Load Forecast for 24 Hours

Hour	۱	۲	۳	۴	۵	۶	۷	۸
Load (MW)	۷۰۰	۷۵۰	۸۵۰	۹۵۰	۱۰۰۰	۱۱۰۰	۱۱۵۰	۱۲۰۰
Hour	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶
Load (MW)	۱۳۰۰	۱۴۰۰	۱۴۵۰	۱۵۰۰	۱۴۰۰	۱۳۰۰	۱۲۰۰	۱۰۵۰
Hour	۱۷	۱۸	۱۹	۲۰	۲۱	۲۲	۲۳	۲۴
Load (MW)	۱۰۰۰	۱۰۵۰	۱۲۰۰	۱۴۰۰	۱۳۰۰	۱۱۰۰	۹۰۰	۸۰۰

Table 6. Part A: Specifications of 10 Generating Units

Unit	**Unit 1**	**Unit 2**	**Unit 3**	**Unit 4**	**Unit 5**
Maximum Generation (MW)	455	455	130	130	162
Minimum Generation (MW)	150	150	20	20	25
a ($/MWh²h)	0.00048	0.00031	0.002	0.00211	0.00398
b ($/MWh)	16.19	17.26	16.6	16.5	19.7
c ($/h)	1000	970	700	680	450
Minimum Operating Time (h)	8	8	5	5	6
Minimum Downtime (h)	8	8	5	5	6
Cold Start Cost ($)	9000	10000	1100	1120	1800
Hot Start Cost ($)	4500	5000	550	560	900
Shutdown Cost ($)	0	0	0	0	0
Initial State (h)	8	-8	-5	-5	-6

Table 7. Part B: Specifications of 10 Generating Units

Unit	**Unit 6**	**Unit 7**	**Unit 8**	**Unit 9**	**Unit 10**
Maximum Generation (MW)	80	85	55	55	55
Minimum Generation (MW)	20	25	10	10	10
a ($/MWh²h)	0.00712	0.00079	0.00413	0.00222	0.000173
b ($/MWh)	22.26	27.24	25.92	27.27	27.79

c ($/h)	370	480	660	665	670
Minimum Operating Time (h)	3	3	1	1	1
Minimum Downtime (h)	3	3	1	1	1
Cold Start Cost ($)	340	520	60	60	60
Hot Start Cost ($)	170	260	30	30	30
Shutdown Cost ($)	0	0	0	0	0
Initial State (h)	-3	-3	-1	-1	-1