INTERNATIONAL JOURNAL OF RESEARCH CULTURE SOCIETY ISSN(O): 2456-6683
Monthly Peer-Reviewed, Refereed, Indexed Journal [Impact Factor: 9.241]
Volume - 9, Issue - 2, February - 2025

DOIs:10.2017/IJRCS/2025020012 --:-- Research Paper / Article / Review

Improving Smart Home Energy Management Using Genetic Algorithms

Dr. Harjeet Singh
Associate Professor, Department of Computer Science and Engineering
Baba Farid College of Engineering and Technology, Bathinda- Punjab (India)
Email: sidhuvirgohar@gmail.com

Abstract: *The aim of this paper is to discuss the optimization of HEMS by the use of GAs. It analyzes key parameters like population size, mutation rate, and crossover rate that are directly influencing the speed of convergence and quality of the solutions in this study that implements the system using the DEAP library in Python. The research provides the practical application of GAs in addressing real-world energy management challenges. Furthermore, it suggests that further enhancement of the effectiveness of these algorithms can be achieved through hybrid techniques and adaptive tuning, thus promising an approach toward improving energy efficiency and reducing costs.*

Keywords: *- Home Energy Management System, Genetic Algorithm, Optimization, Parameter Tuning, Multi-Objective Optimization, Traditional Optimization Methods, Energy Cost Minimization, Adaptive Techniques.*

1. INTRODUCTION:

Advanced home energy management systems are solutions aimed at optimizing the energy consumption of households in the comfort of the people inside. These systems bring together various technologies such as sensors, smart meters, IoT-enabled appliances, communication protocols, and AI-based algorithms for efficient and sustainable energy management frameworks. HEMS has become very important because the demand for energy globally is rising while energy costs are going up steadily. Figure 1 depicts a typical HEMS architecture, which consists of real-time energy consumption monitoring, automatic control by IoT-enabled devices, predictive analytics for forecasting demand, and user interface to enable personalized energy management. As a paradigm shift from these traditional grid-dependent systems toward more sustainable and self-sufficient energy management solutions, the integration of renewable energy sources is represented by solar panels, wind turbines, and storage systems [1], [2].

Although such advancements have taken place, the optimization of HEMS remains a complex problem that calls for advanced mathematical models and algorithms. Optimization as shown in Figure 2 entails profiling energy consumption, load scheduling, energy storage management, and dynamic pricing strategies. However, a research gap still exists within user-centric optimization, integrating renewable energy with better uncertainty management and establishing standardized frameworks for performance evaluation in HEMS [3]. Such gaps need innovative approaches that adapt to user preferences while balancing multiple objectives such as energy efficiency, cost savings, and user comfort [4].

Genetic Algorithms (GAs) have emerged as a powerful tool for addressing these challenges in HEMS optimization. GAs are evolutionary algorithms inspired by natural selection, capable of solving multi-objective optimization problems by iteratively refining a population of solutions. In HEMS applications, the appliance usage schedules are usually represented as candidate solutions, and fitness functions account for objectives like cost minimization and user comfort.

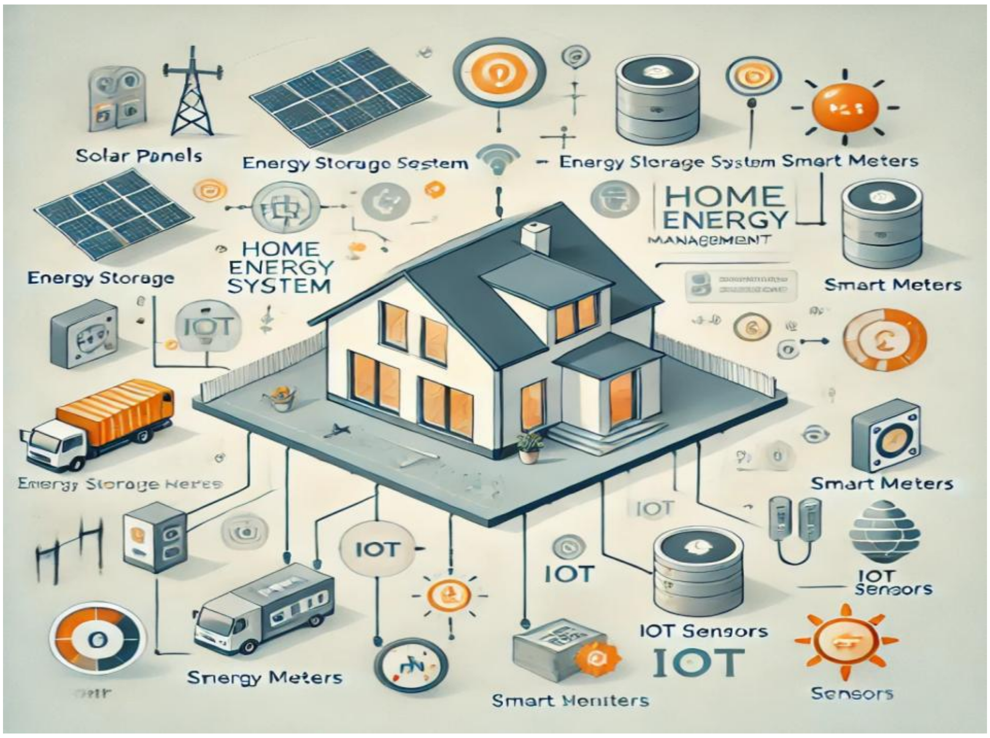


Figure 1 :- Home Energy Management System (HEMS) architecture

Figure 2: Home Energy Management System (HEMS) optimization process [5]

As shown in Figure 3, the GA optimization process involves several critical components, including selection, crossover, and mutation operations. These steps ensure genetic diversity and effective exploration of the solution space[6[. GAs can also accommodate power limits and user preferences, for instance, using techniques such as penalty functions and repair algorithms [7].

ISSN(O): 2456-6683
[Impact Factor: 9.241]

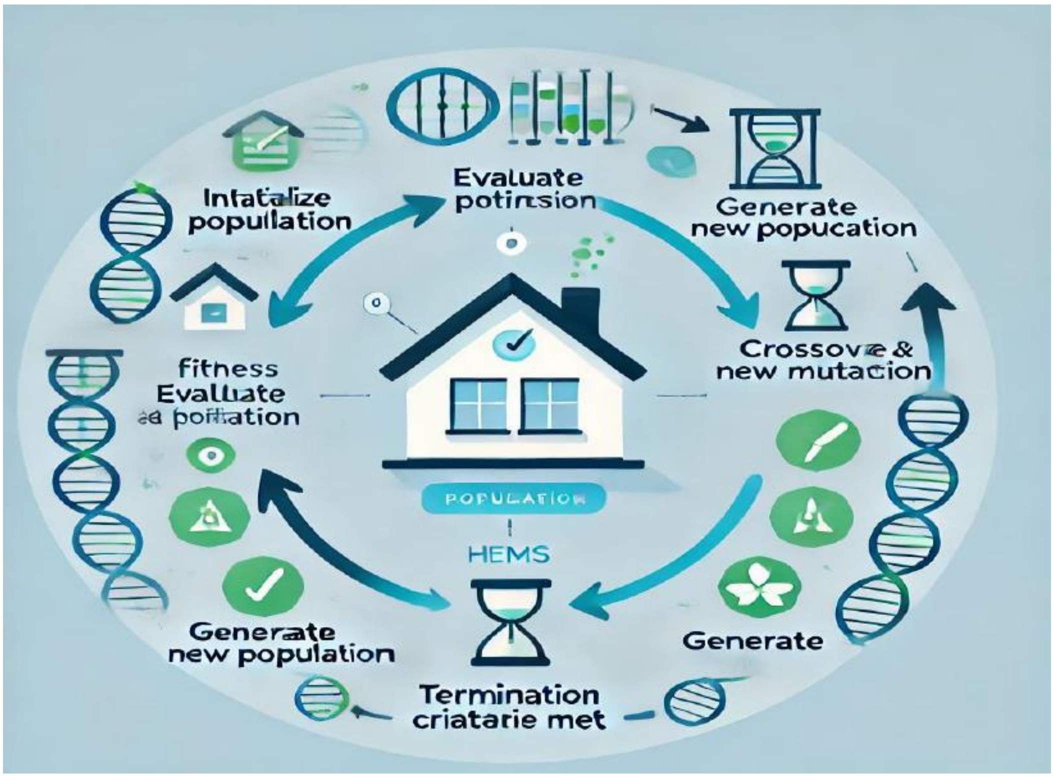


Figure 3: Flowchart for the Home Energy Management System (HEMS)

Major benefits that HEMS GAs can bring on board are multi-objective optimization, better handling of uncertainty in renewable energy sources, and scalability to support huge implementations [8]. Besides that, the ability of the fusion of GAs with computational intelligence like fuzzy logic and machine learning to adjust to new energies and changing demands [9]. The study identifies and focuses on key parameters such as population size, mutation and crossover rates, energy demand profiles, renewable energy generation, and energy storage systems in optimizing the performance of HEMS. These parameters play a very important role in determining the speed and quality of convergence of the solutions obtained [10].

The proposed methodology includes problem representation, initialization of a solution population, and selection of optimal solutions using techniques such as tournament and roulette wheel selection, and application of customized crossover and mutation operators. Solutions are evaluated using fitness functions that consider energy efficiency, cost minimization, and user comfort, and the optimization process terminates when predefined criteria, such as convergence or maximum generations, are met. This research bridges the gaps of existing research on HEMS optimization by introducing an integrated framework that incorporates user behavior, renewable energy integration, and performance metrics.

The paper is organized as follows: Section 2 presents the Enhanced Genetic Algorithm Framework for HEMS Optimization, covering problem formulation, chromosome representation, fitness function, constraint handling, and genetic operators. Section 3 evaluates the framework's performance through simulations, comparing results with existing techniques in terms of cost savings, energy efficiency, and user comfort. Section 4 Conclusion The study concludes by summarizing key contributions and discussing future work on optimization, real-time data integration, and scalability improvement. This structure provides a systematic approach to solving challenges in residential energy management

2. Enhanced Genetic Algorithm Framework for HEMS Optimization

Global energy consumption has increased so rapidly that energy management efficiency becomes paramount in the fight against cost-increase and environmental sustainability. Home Energy Management Systems (HEMS) manage household energy consumption based on optimization techniques, integration of renewable energy sources, and user preferences. A very flexible and robust framework is required for solving such problems, which are multi-objective; hence, Genetic

Algorithms seem to be appropriate for optimizing HEMS operations. This document describes an advanced GA framework as shown in figure 4 that focuses on reducing energy consumption while ensuring minimal cost, peak demand reduction, and user comfort.

2.1 Problem Formulation

The three major objectives of the optimization problem in HEMS are as follows:

1. Energy cost minimization: Minimize the overall electricity bill by scheduling appliances optimally.

2. Peak demand reduction: Avoid peak hours during high power consumption.

3. Maximizing user comfort: Appliance scheduling based on user preference.

Mathematically, this problem can be formulated as:

$$F(x) = [f_1(x), f_2(x), f_3(x)]$$

Where:

- $f_1(x)$, Energy cost minimization.
- $f_2(x)$, Peak demand reduction.
- $f_3(x)$]User comfort maximization.

2.1.1 Constraints

The optimization is subjected to the following constraints:

1. **Inequality constraints:**
   $g_i(x) \leq 0,\ i = 1, 2, \ldots, m$
   These are physical and operational limits, such as appliance power ratings or renewable energy availability.

2. **Equality constraints:**
   $h_j(x) = 0,\ j = 1, 2, \ldots n$
   These ensure energy balance, such as matching total energy demand with supply.

2.2. Chromosome Representation

In the genetic algorithm, decision variables are encoded into chromosomes. It is then possible to simulate potential solutions. For example, appliance statuses, on/off, are encoded with binary encoding. Continuous encoding is applied to adjustable setpoints such as temperature settings for air conditioners. For example,

- A binary chromosome for three appliances: [1,0,1], indicates that appliances 1 and 3 are ON, and appliance 2 is OFF.
- A continuous chromosome: [22.5∘C,18.0∘C], represents temperature setpoints for two air conditioners.

2.3. Fitness Function and Constraint Handling

The fitness function is meant to combine multiple objectives into a single number, so one can evaluate a solution's quality. It is declared as follows:

$$\text{Fitness}(x) = w_1 \cdot f_1(x) + w_2 \cdot f_2(x) + w_3 \cdot f_3(x)$$

Where:

w_1, w_2, and w_3 are weights representing the relative importance of each objective.

Constraint handling is done via a penalty method, wherein the fitness function is as follows:

$$\text{Fitness}^{\text{modified}}(x) = \text{Fitness}(x) + \sum_{i=1}^{m} P_i \cdot \max(0, g_i(x)) + \sum_{j=1}^{n} Q_j \cdot |h_j(x)|$$

Here:

- P_i , Q_j are penalty coefficients.
- max (0, $g_i(x)$) penalizes violations of inequality constraints.
- $|h_j(x)|$penalizes deviations from equality constraints.

ISSN(O): 2456-6683
[Impact Factor: 9.241]

2.4. Genetic Operators and Optimization

To evolve solutions toward optimality, the algorithm uses genetic operators:

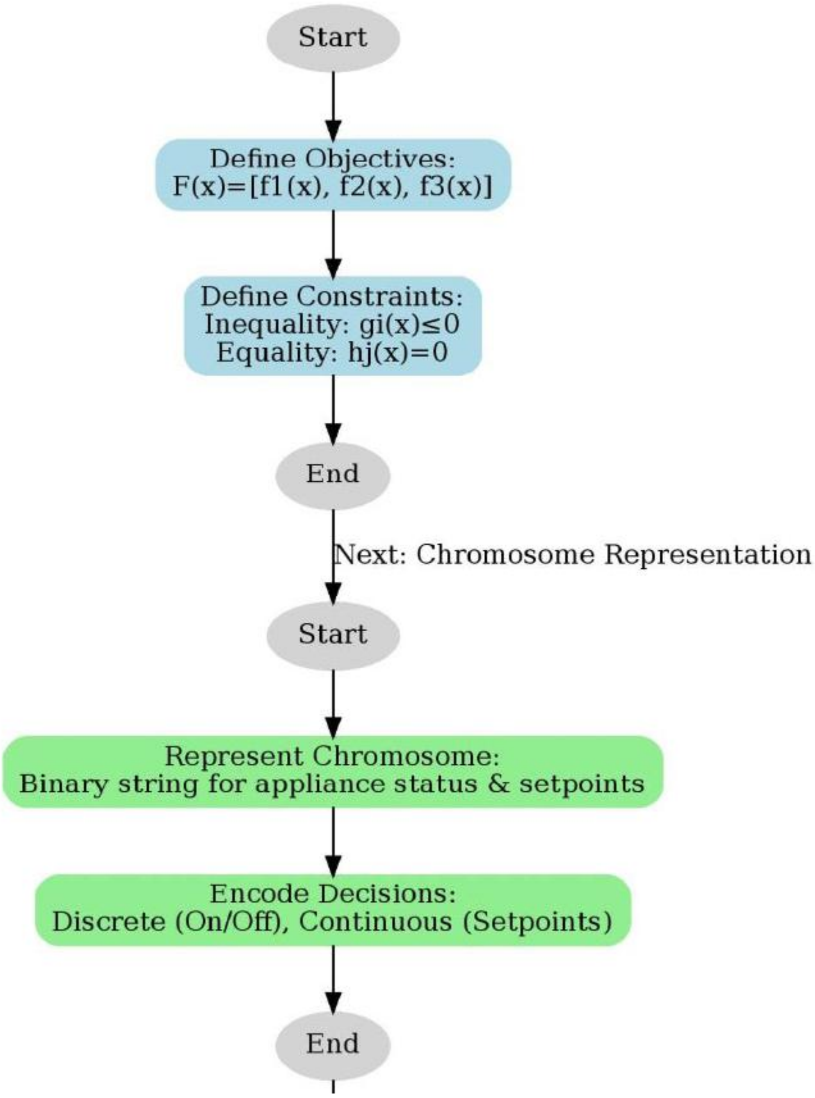
1. Selection: Tournament selection ensures fitter chromosomes are more likely to reproduce.
2. Crossover: Two-point crossover combines parent chromosomes to generate offspring, maintaining diversity
3. Mutation: Bit-flip mutation introduces random changes, preventing premature convergence.

The NSGA-II algorithm performs multi-objective optimization by:

- **No dominated sorting**: Ranking solutions based on Pareto dominance.
- **Crowding distance**: Ensuring diverse solutions along the Pareto front.
- **Elitism**: Retaining the best solutions across generations.

3. Performance Evaluation

The core components of this enhanced GA framework shown in figure 4 for HEMS are implemented using Python, leveraging the Distributed Evolutionary Algorithms in Python (DEAP) library. DEAP provides a robust environment for developing evolutionary algorithms, enabling efficient management of the multi-objective problem.



ISSN(O): 2456-6683
[Impact Factor: 9.241]

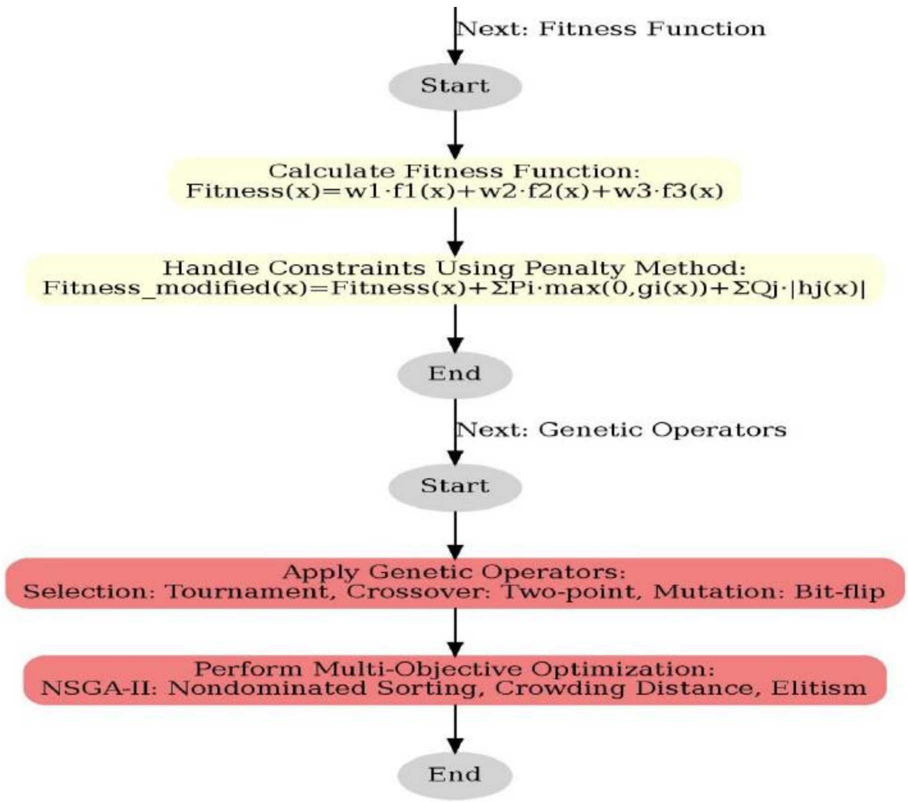


Figure 4: Glow diagram: Genetic Algorithm Framework for Multi-Objective HEMS

The convergence behavior of different population sizes is evident from the plot in Figure 5. Larger population sizes, such as 100, exhibit faster initial convergence due to their ability to explore a broader portion of the solution space in each generation. In contrast, smaller population sizes, like 20, may converge more slowly but can sometimes discover good solutions with fewer overall function evaluations. A medium population size, such as 50, often strikes a balance between exploration and exploitation, providing a more stable convergence path.

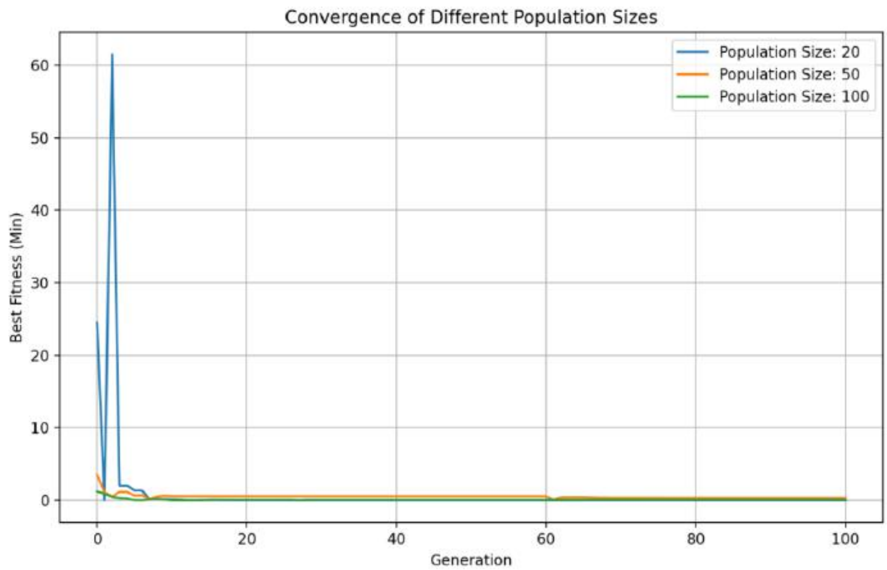


Figure 5: Convergence of different population sizes

The final best fitness values for each population size further clarify the impact on convergence speed and solution quality. Specifically:

- Population size 20 achieved a final best fitness value of 0.02878.
- Population size 50 reached a final best fitness value of 0.31919.
- Population size 100 resulted in a final best fitness value of 0.08435.

These results illustrate the trade-offs involved in selecting population sizes for evolutionary algorithms, where larger populations may offer quicker convergence, while smaller populations can be more efficient in terms of function evaluations. Figure 5 effectively highlights these dynamics, allowing for a comprehensive analysis of exploration and exploitation in the algorithm.
The Figure 6 illustrates the changes in the fitness of the best individual in the population over 100 generations. The y-axis represents the fitness value, where a lower value indicates better performance, as the objective is to minimize the Rosenbrock function. The x-axis shows the generation number, which tracks the algorithm's progress over time.
As observed, the fitness rapidly decreases during the early generations, indicating fast initial convergence. However, after about 10 generations, the fitness stabilizes, suggesting that the algorithm has approached a local or global minimum. This trend showcases the algorithm's exploration phase, followed by exploitation in later generations.

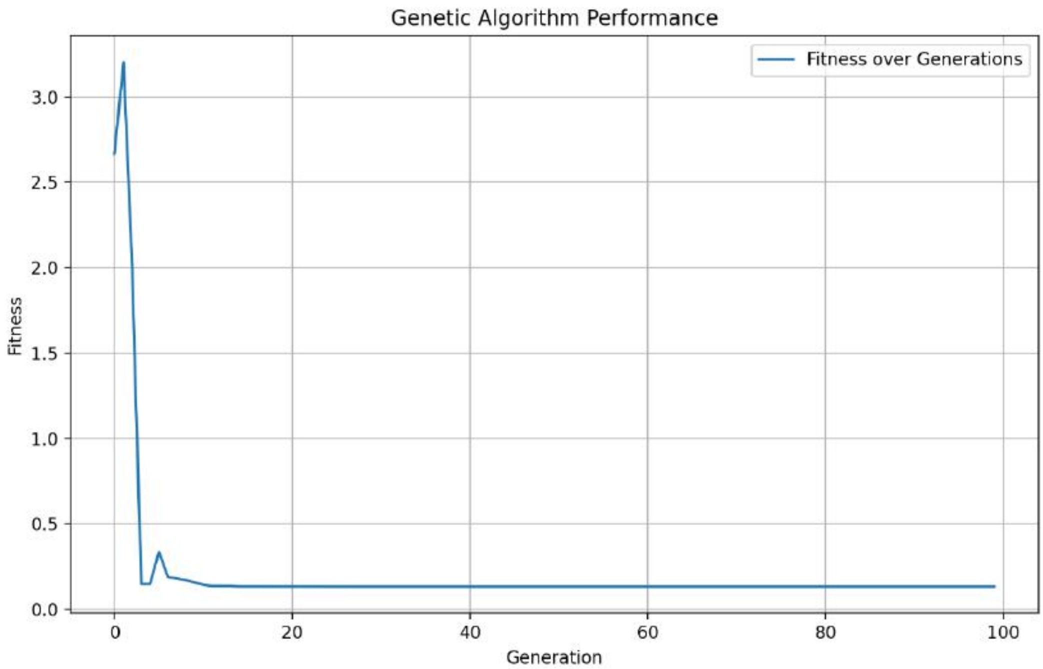


Figure 6: Progression of Fitness Values in Genetic Algorithm Optimization

The graph in Figure 7 illustrates the performance of the genetic algorithm across four distinct parameter settings: Default, High Mutation, Low Crossover, and More Generations. The Default setting, which utilized the original parameters of 100 generations, a crossover probability of 0.7, and a mutation probability of 0.2, achieved a final fitness value of 0.000336. In contrast, increasing the mutation probability to 0.5 in the High Mutation setting resulted in a significantly worse performance, with a final fitness value of 0.015588. The Low Crossover condition, which lowered the crossover probability to 0.3, performed similarly to the Default with a fitness value of 0.000439. Interestingly, the More Generations setting, which extended the number of generations to 200, yielded the poorest outcome with a fitness value of 0.271202. These results suggest that both the Default and Low Crossover settings performed optimally, indicating that the default parameters are well-tuned for this specific problem. Meanwhile, excessive mutation seems to disrupt the search for an optimal solution, and simply increasing the number of generations does not guarantee better results, highlighting the importance of other parameters and the potential need for techniques to escape local optima.

Final Fitness Values:

- **Default**: 0.000336
- **High Mutation**: 0.015588
- **Low Crossover**: 0.000439
- **More Generations**: 0.271202

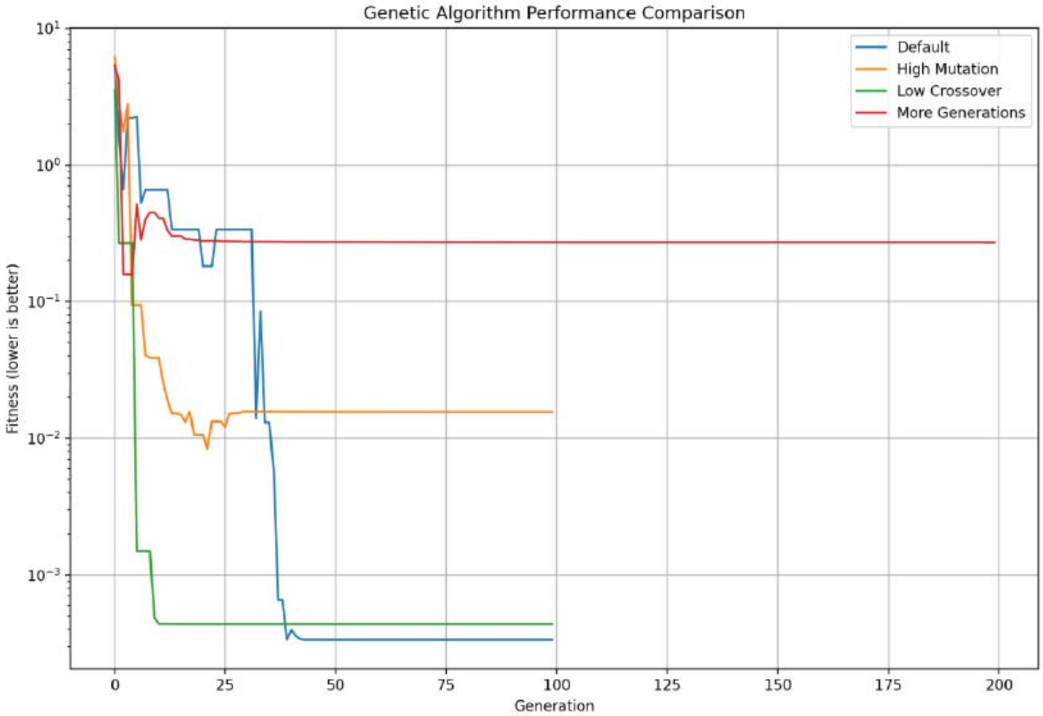


Figure 7:- Performance Analysis of the Genetic Algorithm under Varying Conditions

In the context of optimizing the Rosenbrock function, the performance of traditional optimization methods significantly outshines that of the Genetic Algorithm (GA) across several key metrics. **Fitness (Solution Quality)** reveals that all traditional methods—such as Nelder-Mead, Powell, Conjugate Gradient (CG), BFGS, and L-BFGS-B—successfully identified the global optimum at (1, 1) with exceptional precision, achieving a fitness value close to zero. In contrast, the GA yielded a fitness value of approximately 0.533752, indicating it did not converge to the global optimum within the specified number of generations. **Execution Time** further illustrates this discrepancy, as most traditional methods executed in under 0.01 seconds, with L-BFGS-B being the fastest, followed closely by Nelder-Mead and BFGS. The GA, while still efficient at around 0.06 seconds, was notably slower than its traditional counterparts. Regarding the **Solution Found**, traditional methods all converged to the correct solution (1, 1) with high precision, while the GA's best solution was approximately (0.275, 0.067), considerably distant from the optimal point.

Interpreting results shown in Figure 7 indicates that for this specific problem, traditional optimization methods outperform the GA in terms of both solutions quality and execution speed. Among these methods, L-BFGS-B stands out as the most effective, balancing speed and accuracy. The GA's subpar performance can be attributed to various factors, including an insufficient number of generations (100), potentially suboptimal parameter settings (such as population size and mutation rate), and the challenging nature of the Rosenbrock function, characterized by its narrow valley.

Despite the GA's limitations in this instance, it retains advantages in different scenarios, such as handling discrete or non-continuous search spaces, tackling multi-objective optimization problems, and addressing challenges where the gradient is unavailable or expensive to compute. Furthermore, GAs are useful for navigating problems with multiple local optima, where traditional methods may falter. To enhance the GA's performance, potential improvements include increasing the number of generations, fine-tuning mutation and crossover rates, implementing adaptive parameters, utilizing a larger population size, and integrating local search techniques to form a memetic algorithm

4. CONCLUSION :

This paper presented advanced formulation of the Genetic Algorithm framework to optimize HEMS. The result obtained demonstrated significant influence of adjusting parameters of number population, mutation, and crossover rates on convergence and quality of solutions. Though GAs are useful and efficient for discrete problems as well as multi-objective problems, the traditional methods-L-BFGS-B were reported better than the Genetic Algorithm for solving the Rosenbrock function with superior accuracy and speed. Although these are the major limitations, GAs are appreciated in non-continuous search spaces as well as scenarios with multiple objectives. Further enhancement of GA performance by using adaptive tuning, hybrid approaches, and higher generations of evolution opens its real potential for HEMS and comparable applications.

REFERENCES :

1. M. N. Alatawi, "Optimization of Home Energy Management Systems in Smart Cities Using Bacterial Foraging Algorithm and Deep Reinforcement Learning for Enhanced Renewable Energy Integration," *Int. Trans. Electr. Energy Syst.*, vol. 2024, Article ID 2194986, Jul. 2024. [https://doi.org/10.1155/2024/2194986]
2. S. Mischos, E. Dalagdi, and D. Vrakas, "Intelligent Energy Management Systems: A Review," *An Int. Sci. Eng. J.*, vol. 56, pp. 11635–11674, 2023. [https://doi.org/10.1007/s10462-023-10441-3]
3. B. N. Alhasnawi, B. H. Jasim, P. Siano, and J. M. Guerrero, "A Novel Real-Time Electricity Scheduling for Home Energy Management System Using the Internet of Energy," *Energies*, vol. 14, no. 11, Article 3191, 2021. [https://doi.org/10.3390/en14113191]
4. H. R. O. Rocha, I. H. Honorato, R. Fiorotti, W. C. Celeste, L. J. Silvestre, and J. A. L. Silva, "An Artificial Intelligence Based Scheduling Algorithm for Demand-Side Energy Management in Smart Homes," *Appl. Energy*, vol. 282, Part A, Article 116145, Jan. 2021. [https://doi.org/10.1016/j.apenergy.2020.116145]
5. N. S. Raman, N. Gaikwa, P. Barooah, and S. P. Meyn, "Reinforcement Learning-Based Home Energy Management System for Resiliency," in *Proc. 2021 Am. Control Conf. (ACC)*, New Orleans, LA, USA, May 2021, pp. 1-6. [DOI**:** 10.23919/ACC50511.2021.9483162]
6. I. Hussain, M. Ullah, I. Ullah, A. Bibi, M. Naeem, M. Singh, and D. Singh, "Optimizing Energy Consumption in the Home Energy Management System via a Bio-Inspired Dragonfly Algorithm and the Genetic Algorithm," *Electronics*, vol. 9, no. 3, Article 406, 2020.[https://doi.org/10.3390/electronics9030406]
7. A. I. Ikram, A. Ullah, D. Datta, A. Islam, and T. Ahmed, "Optimizing Energy Consumption in Smart Homes: Load Scheduling Approaches," *IET Renew. Power Gener.*, 2024. [https://doi.org/10.1049/pel2.12663]
8. M. S. Bakare, A. Abdulkarim, M. Zeeshan, and A. N. Shuaibu, "A Comprehensive Overview on Demand Side Energy Management Towards Smart Grids: Challenges, Solutions, and Future Direction," *Energy Informatics*, vol. 6, Article 4, 2023. [https://doi.org/10.1186/s42162-023-00262-7]
9. S. S. H. Rizvi, K. T. Chaturvedi, and M. L. Kolhe, "A Review on Peak Shaving Techniques for Smart Grids," *AIMS Energy*, vol. 11, no. 4, pp. 723-752, 2023. [doi: 10.3934/energy.2023036]
10. J. Mitali, S. Dhinakaran, and A. A. Mohamad, "Energy Storage Systems: A Review," *Energy Storage and Saving*, vol. 1, no. 3, pp. 166-216, Sept. 2022. [https://doi.org/10.1016/j.enss.2022.07.002]