International Journal of Computer Applications (0975 – 8887)
Volume 187 – No.84, February 2026

A Multi-Population Genetic Algorithm for Secure and Efficient Elliptic Curve Parameter Generation

Mohammed H. Alabiech
General Company of Electrical Energy Production – Southern Region,
Ministry of Electricity, Basra, Iraq

ABSTRACT

The field of cryptography has advanced rapidly in recent years, with Elliptic Curve Cryptography (ECC) emerging as one of the most efficient methods for ensuring both security and computational efficiency. The strength of this encryption technique is largely determined by the mathematical properties of the Elliptic Curves (EC), which are governed by the constants defining its structure. This study explores the use of a Genetic Algorithm (GA)—an evolutionary artificial intelligence technique—to determine optimal values for the constants of EC, aiming to maximize the number of valid points over a finite field. This approach highlights the feasibility of applying intelligent optimization techniques to complex mathematical challenges in cryptographic system design. A prototype was implemented to simulate the process and assess the GA's performance in identifying effective solutions. Preliminary results suggest that the GA offers a viable alternative to traditional search methods, enabling more efficient exploration of cryptographic parameters. This could contribute to designing more efficient EC for cryptographic applications and deepen our theoretical and practical understanding of EC construction.

General Terms

Security, Optimization, Artificial Intelligence, Performance.

Keywords

Elliptic Curve Cryptography, Genetic Algorithm, Base Point.

1. INTRODUCTION

Elliptic Curve Cryptography (ECC) has emerged as a key component of modern information security, offering superior efficiency compared to traditional public-key algorithms such as RSA and DSA [1]. The security of ECC is primarily determined by the computational difficulty of solving the Discrete Logarithm Problem (DLP) [2]. This paper provides a brief overview of EC key sizes and their corresponding security levels, in comparison with the RSA cryptosystem.

The Table 1. below explains the security strength of the ECC to RSA from where of key size and period in Million Instructions per Second (MIPS).

Table 1. Secret and public key sizes with equivalent security levels

ECC (bit)	RSA (bit)	Time to be break in MIPS
106	512	10^4
160	1024	10^{11}
210	2048	10^{20}
600	21000	10^{78}

ECC is based on the mathematical structure of points on an EC over a finite field, where the complexity of point operations contributes significantly to encryption strength [3].

A solid understanding of the underlying mathematics of EC is essential to implement ECC effectively. Despite their name, EC are not related to ellipses; they originate from specific types of cubic equations [4]. The standard equation for an EC over the real numbers is given by [4-8]:

$$y^2 = x^3 + ax + b \qquad (1)$$

whereas a, b are real number and satisfying the condition

$$4a^3 + 27b^2 \neq 0 \qquad (2)$$

and x, y assume value in the real number.

The prime curve over Z_p (where $p > 3$) is defined by the cubic equation shown below:

$$y^2 \bmod p = (x^3 + ax + b) \bmod p \qquad (3)$$

Where

$$(4a^3 + 27b^2) \bmod p \neq 0 \qquad (4)$$

The main contributions of this paper can be summarized as follows:

1. Proposing a multi-population steady-state GA framework for elliptic curve parameter optimization.
2. Providing a comparative experimental evaluation against the traditional exhaustive search method.
3. Demonstrating scalability advantages for large prime numbers.
4. Validating the optimization results with respect to Hasse's theoretical bounds.

2. RELATED WORKS

Numerous studies have investigated the unique mathematical properties of EC for enhancing encryption and authentication in diverse security applications. Neal Koblitz and Victor Miller independently introduced the use of ECC in 1985. Their work laid the foundation for ECC as an efficient alternative to traditional public-key cryptosystems, providing enhanced security with significantly smaller key sizes[4]. Numerous studies have explored the use of Genetic Algorithms (GAs) to refine ECC parameters, aiming to enhance security while optimizing computational efficiency. For instance, Kumar and Singh proposed a GA-based method for optimizing EC parameters, aiming to enhance cryptographic robustness and reduce computational overhead. Their methodology showed greater resilience against specific cryptanalytic attacks compared to conventional fixed-parameter curves[9]. The concept of multi-population GAs, often referred to as the island model, has been widely explored to improve both stability and convergence in evolutionary optimization. In 2019, Zhang et al. proposed a framework that preserves diversity among subpopulations, preventing premature convergence. Their approach has demonstrated effectiveness in various optimization challenges, particularly in fine-tuning cryptographic parameters[10]. Several evolutionary techniques—such as Particle Swarm Optimization (PSO) and Differential Evolution (DE)—have also been applied to optimize EC parameters.

However, Patel and Mehta conducted a comparative analysis and found that Gas provide a more effective balance between exploration and exploitation in this context. Their findings reinforce the potential of multi-population GA in enhancing parameter robustness and search efficiency[11]. Azam et al. has presented a new approach to digital watermarking by leveraging GAs and ECC. The study aims to enhance watermarking security and efficiency by integrating evolutionary optimization techniques with strong cryptographic principles. GAs enable adaptive optimization, while EC provide robust encryption, making the scheme highly resilient against attacks while maintaining image fidelity. This method is particularly useful for protecting intellectual property and ensuring data integrity in digital media[12].

3. BASE POINT

In general, EC based protocol, it is essential to yield a base point of order n, preferably n is a large prime, and $G \neq \infty$ such that $nG = \infty$. As the cryptographic strength of EC relies on n, it is preferable to maximize n[6].

The order of G in should check is not a smooth integer (small integer takes G to point at infinity)[13]. Base point choice in ECC is the main procedure for its security[14]. The number of point and order of each point depend on the selection of EC parameters (p, a, b,G) over F_p. The selection of any point on EC as a base point depends on the order of this point; therefore, to select base point G, first; determine the order of all points on EC over finite field. Since, the large order of base point give a large range to select the private key; hence, the point that has the maximum order is the best point to select it as a base point. The order of any point $n \leq \#E(F_p)$, if $n = \#E(F_p)$ then the point is generator, that mean the addition of point to itself give all points on EC (the point generates all other points on EC), if there is one or more generators in group of points then, the group of points is called *cyclic group*[6].

The cyclic groups of points are the best groups of points that can select one of its generators as a base point. If all points of cyclic group are generators, then, any point of this group can be selected as a base point, and the key can generate any value of private key.

Algorithm 1. describes the steps involved in selecting the best base point, providing a clear and structured overview of the process[15].

Algorithm 1 Selection of the best base point on EC

```
Input: $p$
Output: $n$
 1:  Select $a, b \in [0, p-1]$, where $(4a^3 + 27b^2) \bmod p \neq 0$
 2:  Determine all the point $F_p$.
 3:  Select one of the points as a base point $G(x,y)$
 4:  if $y$=0 then $n \leftarrow 1$
 5:    else Doubling of $G$.
 6:      repeat
 7:        determine the solution of $(kG+G)$ for $k$=2,3…
 8:      until $(kG+G) \longrightarrow \infty$
 9:      $n \leftarrow k+1$
10:  end if
11:  repeat the steps 3-10 to the other point
12:  return the point that has maximum order($n$)
```

4. GENETIC ALGORITHM

Genetic algorithm (GA) is a general tool for search and optimization based on the techniques of natural selection.

Genetic Algorithms (GAs) provide solutions to optimization problems by employing mechanisms inspired by natural evolution[16]. It not only provides an effective search technique but can also be utilized as an efficient optimization tool.

Potential solutions (i.e., population which contains a set of chromosomes) generated randomly to a problem and then applying different operators, such as: selection, crossover , and mutation in order to achieve increasingly better results[17, 18]. A GA begins with an initial population of chromosomes, each representing a potential solution. In every generation, selected chromosomes are used to generate a new population through variation operators. This is due to a desire that the new population will be more useful than the old population. Chromosomes which are chosen to form new offspring's are taken in accordance with their fitness (i.e., the more appropriate chromosomes are the more chances they have to re-procreate). GA finishes when either an acceptable fitness level is achieved or a given number of generations is reached.

The general algorithmic structure of GA is explained in Algorithm 2. below[16]:

Algorithm 2 The General Algorithm for GA

```
1:  Initialize population;
2:  repeat
3:     Evaluation;
4:     Reproduction;
5:     Crossover;
6:     Mutation;
7:  until (terminus conditions are met);
```

4.1 Population and Initialization

A population is a set of chromosomes (i.e., solutions for a .given problem). The number of chromosomes is named Population Size which is a user-defined. Each chromosome consists of a set of genes, and the initial population is typically generated randomly[19].

4.2 Evaluation

After the population is initialized or a new offspring is created, it is necessary to compute a fitness value for a chromosome(s) by using a suitable fitness function. The fitness function returns a value that reflects the quality of a given chromosome. Selecting a suitable function depends on the applications and plays a crucial role to success the algorithm[20].

4.3 Selection

In the selection operator, a set of chromosomes are chosen as parents depends on their fitness values to form a breeding population. There are two main selection mechanisms[16]:

1. *Roulette wheel selection*: It is a classical choice mechanism. The better the chromosomes are, the more chances to be taken they have.
2. *Tournament selection*: A set of chromosomes which are randomly chosen from the population, the one with the best fitness value being as a parent. This process is repeated as parent must be selected.

4.4 Crossover

A crossover operator combines the parents with crossover probability Pc to form offspring(s). There are different crossover operators in literature[21].

The main three popular crossover operators are:

1. *One-point crossover (1x)*: Where the two parents join at randomly one crossover point and interchange the parent genes after this point to produce two offsprings.
2. *Two-point crossover (2x)*: Where the two parents join at randomly two crossover points and interchange the parent genes between these two points to produce two offsprings.
3. *Uniform crossover (ux):* Where a coin toss is performed at each corresponding gene to determine whether or not an interchange of two genes pass at that position.

4.5 Mutation

Mutation is the process of randomly altering one or more genes within a chromosome, selected with a probability Pm to produce a new chromosome[22].

4.6 Terminus Conditions

The steps 3-6 (which are called a generation) in the Algorithm 2. are repeated until one or more termination conditions are met. The popular conditions are: (i) reach a user-define number of generations; (ii) the population is identical; (iii) the fitness function achieves a specific threshold; and (iv) combination of the above[18].

5. SELECT *a*,*b* PARAMETERS (PROPOSED METHOD)

As a process to select *a*,*b* with condition by using Equation 4, now; what the values of these parameters are most effective? the answer of this question is the parameters which generate maximum number of #E(F_p) when compensate the *a* and *b* in Equation 3, in order to increase the range to select private keys, for this reason, an approach is developed to select the best parameters using GAs. The traditional method for select parameters, for example, is very costly. In order to obtain the curve with most effective parameters with maximum order of a group. Now, the traditional method has been clarifies in the Algorithm 3.

Algorithm 3 The Traditional Algorithm

Input: *p* is prime number.
Output: *a* and *b* are the parameters of EC, *n* is order of a group.

```
 1:  set i ← 1
 2:  for $a_1$ = 1 to $p$-1
 3:      for $b1$ = 1 to $p$-1
 4:        if $(4a_1^3 + 27b_1^2) \bmod p \neq 0$ then
 5:            set $c1$ ← 1
 6:          for $x$ = 1 to $p$-1
 7:            for $y$ = 1 to $p$-1
 8:              if $y^2 \bmod p = (x^3 + a_1x + b_1) \bmod p$ then
 9:                $c_1$ ← $c1$ +1
10:            end if
11:            end for
12:          end for
13:          if ($c_1$ is prime) & ($c_1 \neq p$) then
14:            $a_i$ ← $a_1$, $b_i$ ← $b_1$, $n_i$ ←$c_1$
15:            i ← i+1
16:          end if
17:        end if
18:      end for
19:  end for
20:  return $a$ and $b$ with maximum $n$
```

In the current method, the well-known GA that is called steady state GA (ssGA) is used. The ssGA algorithm is given in the Algorithm 4.

Algorithm 4 The Basic Algorithm for ssGA

popsize	size of population.
population	population of chromosomes.
pc, pm	probability of crossover and mutation, respectively.

```
 1:  Initialization (population, popsize)
 2:   Evaluation (population)
 3:  repeat
 4:    parents ← Selection (population, popsize)
 5:    offspring ← Crossover (pc, parents)
 6:    offspring ← Mutation (pm, offspring)
 7:     Evaluation (offspring)
 8:     Replacement (population, popsize, offspring)
 9:    until (termination conditions are met)
10:  return the best individual in the population
```

New technique uses GA to decrease the costs of selecting parameters, as explained in the following stages:

5.1 Initialization

The first stage of the ssGA is the initialization of population; the population contains here 50 chromosomes.
Each chromosome consists of two integer genes that represent the parameters *a* and *b* of the ECC Equation 3.
The genes are generated randomly that satisfy the Equation 4.
In order to avoid weak curve, the *n* should be prime number and is not equal *p*.

5.2 Evaluation

In order to evaluate each chromosome (C), the current work uses the following fitness function: $f(C) = n$.

5.3 Selection

The selection scheme is to select two parents as follows: the first parent is the fittest individual in the population, while the second one is selected randomly.

5.4 Crossover

The crossover operator is *1x* with crossover probability Pc equal to 0.8.

a_1	b_1
12	34

a_2	b_2
123	1

$\longrightarrow$

a_1	b_2
12	1

a_2	b_1
123	23

5.5 Mutation

The mutation operator is 1m with mutation probability Pm equal to 0.2. The new chromosome must satisfy the Equation 4.

a_1	$\boldsymbol{b_1}$
12	**1**

$\longrightarrow$

a_1	$\boldsymbol{b_1}$
12	**520**

5.6 Replacement

The replacement scheme works as follow: the new offsprings replace the worst two individuals in the population if these offsprings are fitter than the worst two individuals.

5.7 Terminus Conditions

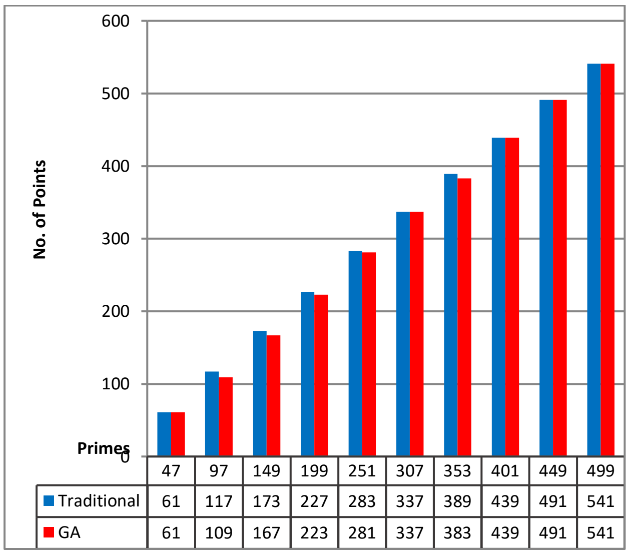
The generation is repeated for 500 generations.

6. EXPERIMENTAL ENVIRONMENT

The simulation environment for this study was configured using MATLAB R2014a, executed on a 32-bit system featuring an Intel Core i5 processor with a clock speed of 3.16 GHz and 4 GB of RAM.

7. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

The proposed technique uses the ssGA in order to obtain the most effective parameters: *a* and *b* with maximum order of a group. So, the study implements the algorithm for some prime numbers and compares the result with the traditional method and noticing the percentage of ssGA results and processing time. Figure 1. shows the result of the proposed technique after 10 runs. As can be seen in the results presented in Figure 1. both GA and the traditional technique achieved very nearly similar results (i.e., the number of points) despite the fact that GA started with random population for each run. In order to compare the behavior of the GA, it compared with the traditional once, a reasonable small prime numbers are taken in the current experimental, since the traditional technique takes a long time for execution (e.g., weeks) if the prime number is large.



Primes	47	97	149	199	251	307	353	401	449	499
🟦 Traditional	61	117	173	227	283	337	389	439	491	541
🟥 GA	61	109	167	223	281	337	383	439	491	541

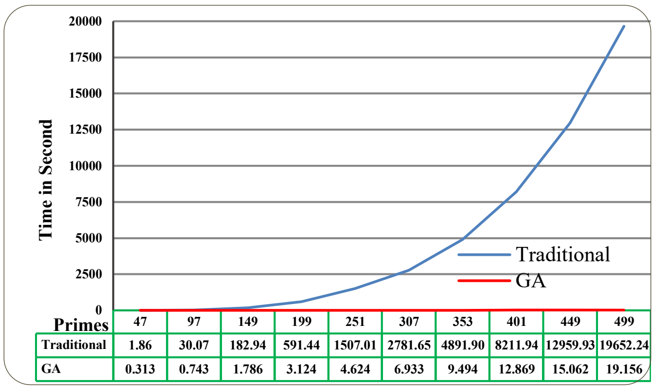
Ratio	100	93.2	96.5	98.2	99.3	100	98.5	100	100	100

Fig 1: The drawing of No. of points (Traditional Vs. GA).

For each prime number, the algorithm was executed over 10 independent runs, and the average number of points along with the processing time were calculated. The results demonstrate minimal variation between runs, confirming the stability and robustness of the proposed approach despite random initialization.

As illustrated in Figure 2., the GA demonstrates significantly better performance in terms of processing time compared to the traditional method. The GA curve begins at 0.313 seconds when the prime number is 47 and rises with very small increments when the prime numbers are increase to reached (19.156 sec.), (i.e.,when prime number equal to 499). In contrast, the traditional method starts at 1.86 seconds (i.e., when prime number equal to 47) and rises exponentially with the prime numbers are increase reached until (19652.24 sec.),

(i.e., when prime number equal to 499).



Primes	47	97	149	199	251	307	353	401	449	499
Traditional	1.86	30.07	182.94	591.44	1507.01	2781.65	4891.90	8211.94	12959.93	19652.24
GA	0.313	0.743	1.786	3.124	4.624	6.933	9.494	12.869	15.062	19.156

Fig. 2: Processing time of algorithms traditional, GA.

To provide a deeper quantitative evaluation, the average processing time over 10 independent runs was computed for each prime number. The results indicate that the GA achieves an average speedup exceeding 95% compared to the traditional exhaustive search method. This significant improvement can be attributed to the guided search mechanism of the GA, which restricts exploration to promising regions of the search space rather than evaluating all possible parameter combinations.

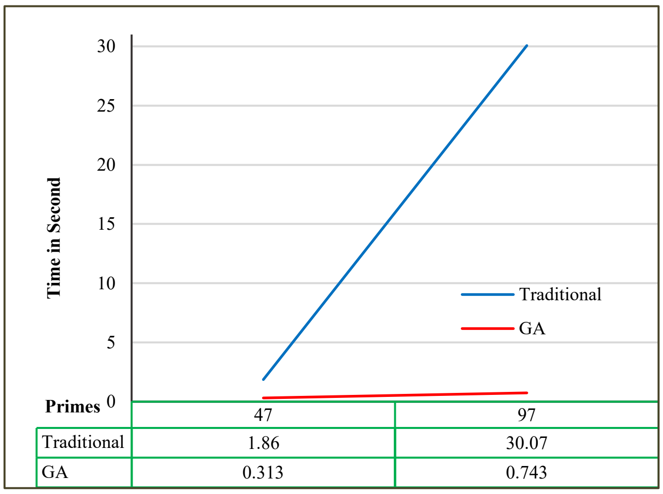
Table 2: Performance comparison between traditional method and GA

Prime	Traditional Time (Sec.)	GA Time (Sec.)	Improvement (%)
47	1.86	0.313	83.2
97	30.07	0.743	97.5
149	182.94	1.786	99.0
199	591.44	3.124	99.5
251	1507.01	4.624	99.7
307	2781.65	6.933	99.8
353	4891.9	9.494	99.8
401	8211.94	12.869	99.8
449	12959.93	15.062	99.9
499	19652.24	19.156	99.9

As shown in Table 2, the percentage of improvement increases significantly as the prime number grows. This indicates that the computational complexity of the traditional method increases exponentially, while the GA maintains a near-linear growth pattern. Such behavior highlights the scalability advantage of the proposed method.

Furthermore, the convergence behavior of the GA demonstrates stability across multiple runs, indicating robustness against random initialization effects. The near-maximal values of #E(F_p) obtained in Table 3 confirm that the proposed method effectively explores the solution space while maintaining cryptographic validity constraints.
the results also align with Hasse's theorem bounds, showing that the GA consistently approaches the upper limit of the theoretical interval, thereby validating the optimization capability of the proposed approach.

And to illustrate the drawing more, Figure 3. illustrates the processing time of traditional and GA for two prime numbers (47 and 97).



Primes	47	97
Traditional	1.86	30.07
GA	0.313	0.743

Fig. 3: Processing time of algorithms traditional, GA for numbers (47 and 97).

In order to check the effectiveness of the GA with large prime numbers (e.g., more than 1000), the GA is tested on 13 prime numbers in the range (1009-13681) and compare its results with Hasse theorem that represented in the following bound:

$$p + 1 - 2\sqrt{p} \leq \#E(Fp) \leq p + 1 + 2\sqrt{p}$$

The performance of the GA for the large prime numbers is given in Table 3.

Table 3 The number of points of Hasse theorem and GA.

Prime	Number of points	
	Hasse theorem	GA
1009	(946-1073)	1051
2099	(2008-2191)	2143
3257	(3143-3372)	3361
4219	(4090-4349)	4349
5101	(4959-5244)	5227
6143	(5987-6300)	6273
7919	(7742-8097)	7877
8779	(8592-8967)	8941
9349	(9156-9543)	9511
10459	(10254-10664)	10601
11173	(10962-11384)	11251
12641	(12417-12866)	12853
13681	(13448-13915)	13907

The experimental results presented in Table 2 demonstrate that the values obtained by the GA consistently approach the upper bound of Hasse's theorem. This observation confirms that the proposed optimization strategy effectively searches within the valid mathematical domain while preserving cryptographic constraints. The ability to approach the theoretical maximum without exhaustive enumeration further validates the efficiency of the evolutionary search process.

Overall, the current findings of the GA experiments are encouraging and provide us with evidence that GA works effectively for the problem under study. So, this technique will be used in the selection of *a*, *b*, and *n* for all schemes which are illustrated in the current study.

8. CONCLUSION AND FUTURE WORKS

This study presented a multi-population Genetic Algorithm for optimizing elliptic curve parameters with the objective of maximizing the number of valid points over finite fields. The experimental evaluation confirmed that the proposed approach significantly reduces computational complexity compared to the traditional exhaustive method while maintaining cryptographic robustness. The results demonstrated not only computational efficiency but also strong convergence behavior and stability across multiple runs. Furthermore, the proximity of the obtained solutions to the theoretical Hasse bounds validates the effectiveness of the evolutionary optimization strategy. The integration of intelligent search mechanisms into elliptic curve parameter generation provides a promising direction for future cryptographic system design, particularly in large-scale and high-security environments.

Finally, the suggestions below are provided for the future work to improve the proposed scheme:

1. The study speculates that further work, by selecting an appropriate population size based on the prime number, would improve the current system's results and make it more realistic.
2. To further reduce processing time, implementing parallel processing using a Graphics Processing Unit (GPU) presents an effective solution. GPUs, particularly those based on NVIDIA's GeForce architecture, can significantly accelerate large-scale computations.

9. REFERENCES

[1] Banerjee, A. and U. Banerjee, 2024. A High-Performance Curve25519 and Curve448 Unified Elliptic Curve Cryptography Accelerator. In Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC). IEEE Boston Section.

[2] K. Vardhan, and V. Jain, 2025. Enhanced Secure File Transfer: A Comparative Analysis of Elliptic Curve Cryptography vs. RSA. In Proceedings of the International Conference on Advanced Computing Technologies (ICoACT). IEEE Madras Section, Sivakasi.

[3] S. Manickam and D. Kesavaraja, 2016. Secure Multi Server Authentication System Using Elliptic Curve Digital Signature. In Proceedings of the International Conference on Circuit, Power and Computing Technologies (ICCPCT). Nagercoil, IEEE.

[4] W. Stallings, 2011. Cryptography and Network Security Principles and Practice. Pearson, Fifth Edition.

[5] C. Pelz, 2010. Understanding Cryptography. Springer-Verlag Berlin Heidelberg.

[6] D. Hankerson and A. Menezes, 2004. Guide to Elliptic Curve Cryptography. Springer-Verlag New York, Inc.

[7] M. H. Alabiech, and H. M. Al-Mashhadi, "Hybrid Menezes Vanstone-ElGamal ECC Algorithm, " Iraqi Journal of Science, vol. 66, no. 3, pp. 1300-1310. 2025.

[8] H. M. Al-Mashhadi, and M. H. Alabiech, "Symmetric ECC with Variable Key using Chaotic Map, " International Journal of Computer Science Issues (IJCSI), vol. 14, no.6, pp. 24-28, 2017.

[9] R. Kumar and A. Singh, "Optimization of Elliptic Curve Parameters Using Genetic Algorithms, " Journal of Cryptographic Engineering, Springer Science + Business Media,. vol. 8, no2, pp. 121-133, 2018.

[10] L. Zhang, H. Wang, and Y. Li, "Multi-Population Genetic Algorithms for Robust Optimization, " Applied Soft Computing, Elsevier, vol. 77, pp. 430-441. 2019.

[11] D. Patel, and S. Mehta, "Comparative Study of Evolutionary Algorithms for Elliptic Curve Parameter Optimization, " Journal of Computer Mathematics, vol. 97, no. 5, pp. 1050-1064, 2020.

[12] N. Azam, T. Haider, and U. Hayat, "An Optimized Watermarking Scheme based on Genetic Algorithm and Elliptic Curve, " Swarm and Evolutionary Computation, vol. 91, no. 5, pp. 1-16, 2024.

[13] Koblitz, N., "Elliptic Curve Cryptosystems," Mathematics of computation, vol. 48, no. 177, pp. 203-209, 1987.

[14] L. Chen, D. Moody, and A. Regenscheid, 2019. Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters, National Institute of Standards and Technology, Tech. Rep.

[15] G. Jwad, 2007. Design and Implementation of Elliptic and Hyper Elliptic Curve Cryptosystems with Improved Performance, PhD Thesis, College of Engineering, University of Basrah.

[16] M. Alabbas, 2013. Textual Entailment for Modern Standard Arabic, PhD Thesis, School of Computer Science, University of Manchester: Manchester, UK.

[17] B. Meniz and F. Tiryaki, "Genetic Algorithm Optimization with Selection Operator Decider, " Arabian Journal for Science and Engineering, Springer, vol. 50, no. 10, pp. 6931–6941, 2025.

[18] S. Sivanandam and S. Deepa, 2007. Introduction to Genetic Algorithms. Springer Science & Business Media.

[19] E. Kocyigit, M. Korkmaz, O. Sahingoz, and B. Diri "Enhanced Feature Selection Using Genetic Algorithm for Machine-Learning-based Phishing URL Detection, " Applied Sciences, vol. 14, no. 14, pp. 1-21, 2024.

[20] T. Avdeenko, and K. Serdyukov, "Modified Evolutionary Test Data Generation Algorithm Based on Dynamic Change in Fitness Function Weights, " Engineering Proceedings, vol. 33, no. 23, pp. 1-9, 2023.

[21] A. Uzunoglu, C. Gahm, and A. Tuma, "Machine Learning based Algorithm Selection and Genetic Algorithms for Serial-batch Scheduling, " Computers & Operations Research, Elsevier, vol. 173, pp. 1-20, 2025.

[22] B. Miller, 1997. Noise, Sampling, and Efficient Genetic Algorithms, PhD Thesis, University of Illinois at Urbana-Champaign.