Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

Hybrid between Genetic Algorithm and Artificial Bee Colony for Key Generation Purpose

Ali AbdulKadhim Taher[a] , Suhad Malalla Kadhim[b]

[a] *Computer Science, University of Technology , Iraq.Email: 0110114@student.uotechnology.edu.iq*

[b] *Computer Science, University of Technology , Iraq.Email: suhad_malalla@yahoo.com*

ARTICLE INFO

Article history:
Received: 06 /08/2019
Rrevised form: 25 /08/2019
Accepted : 09 /09/2019
Available online: 01 /10/2019

Keywords:

Genetic algorithm,
Artificial Bee Colony,
Key generation,
Cryptography.

ABSTRACT

Cryptography is a significant tool for guarantee a protected and secured data. In asymmetric key algorithms, the key generation plays an important role for applied the security. This research focus on enhance genetic algorithm's population by using artificial bee colony to generate keys that are unique and random.

The five statistical tests (Frequency test (monobit test), Serial test (two-bit test), Poker test, runs test and autocorrelation test) are used to compute the fitness.

The programming language Pythom3 was used in programming the proposed method. The proposed algorithm is acquired higher quality solutions within a reasonable computational time compared to traditional genetic algorithm.

DOI : 10.29304/jqcm.2019.11.4.627

1 . Introduction

Random numbers are numbers generated by an operation, which unpredictable it's outcome, and which cannot be sequentially reliably reproduced. Random numbers are necessary for a variety of applications such as cryptography, computer simulation, statistical sampling, mathematical, engineering, and other areas were producing an unpredictable result is desirable. For instance, popular cryptosystems used keys that should be produced in a random mode [1].

Artificial bee colony (ABC) algorithm is proposed by Karaboga in 2005, which is simple in concept and requires very few initialization parameters to adjust. It has properties of fast convergence speed, good quality of solutions and good robustness, etc. but it also has the disadvantage of premature convergence, and is easy to fall into local optimum[2].

Corresponding author Ali AbdulKadhim Taher

Email addresses: 0110114@student.uotechnology.edu.iq

Communicated by Qusuay Hatim Egaar

Genetic algorithm (GA) has good global search ability through its operation strategies, but it has a slow convergence speed[3].
This paper will use a hybrid between GA and ABC to avoid the disadvantages at each one and benefit from their advantages only for random key generation purpose and the five basic tests are used to compute the fitness.

2. Related work

The related work in this paper is divided into two categories:

A: The following are the related works for random key generation by using intelligent optimization algorithms:

1-Jawaid, in 2015[4]. This paper used the GA is used for key generation . The final key is selected based on the autocorrelation value and thus it is as random and unique as possible. the Data Encryption Standard cipher program is used for verification and validation. The proposed solution contains three rounds and can be repeated N times. After the initial population generation, the autocorrelation implemented on the generated population to check for randomness. Next, after Crossover the autocorrelation implemented on the population generated. Similarly, after Mutation autocorrelation is implemented on the obtained population. In the result set, three sets of the population obtained from each step and select the population having the autocorrelation value nearest to zero which gets stored in the repository. Regardless of this, the key is created in a small amount of time. This is a great advantage because the computational time to produce the key is less than encrypting the data using DES.
2-J.Sai Geetha, in 2015[5]. uses Artificial Bee Colony for Random Number Generator. The Run test(up and down) method and also(above and below) is carried out . The key generated are random, non-repeating and strong with large linear complexity. so, keys produced are also safe from related key attacks. Keys generated also successfully passed run test and the poker test.
3-Fadheela Sabri Abu-Almash, in 2016[1]. The genetic algorithm also used to propose a new approach to generates keys. The Five basic tests applied as a fitness function in this experiment for each chromosome. The results proved that the competence of genetic algorithm for key generation. The optimal representation for Genetic Algorithm may be seen by initial population, size of population, crossover and mutation. The large population size, little mutation probability and a high crossover probability.
4-Majedah Alkharji, in 2018[6]. A method to use GA to generate keys for the fully homomorphic encryption scheme(FHE) is described and its effectiveness is examined. GA is applying to get the maximum randomness for the 4096-bits RSA key thus guaranteeing more security. Results describe the efficiency of the proposed method under two different scenarios. First, the time taken to generate a key from 10000 iterations with 64 populations and 30 mutation operations is 3.5760 milliseconds. Second, the test performed to check the nature of randomness shows that the final population has '0' correlation with the initial one. Both results prove that the key generated by GA are unique, strong, non-repeating, high-quality random ,and will enhance the security of FHE schemes, making it more difficult for the cryptanalysts to break the data.

B: The following are the related works for enhancement of GA:

1-Yong Deng in 2015[7]. The population of the genetic algorithm was improved in this work by developing a new initial population strategy to solve the TSP. In this strategy, the travel route is restructured by reconnecting each block. This strategy is called K-means Initial Population strategy (KIP), this strategy was tested in fourteen different TSPLIB models and two other methods (Greedy initial population GIP and Randomly initial population RIP) are used for the experiment. The proposed algorithm can reduce the best line value, and average error value compared to RIP and GIP in the same iterations, and in the same running time.
2-Ajay Shrestha in 2016[8]. An algorithm for the refinement of nature-inspired by the crossover is proposed by authors using an untapped idea of mitochondrial DNA (mtDNA). MtDNA is a small subset of the general DNA. Distinguishes itself by its complete inheritance from the female while the rest of the DNA is inherited equally from both parents. This property can be used to identify members with similar genes and restrict their exchange. Apply a low concept in solving TSP and to train Neural Network for function approximation. The results show that the benefit of this concept produces a relative improvement in the quality of GA's improvement.

3-Abid Hussain in 2017[9]. The authors proposed a new crossover operator to resolve the TSP. The new approach is linked to path representation which is the most natural way to represent a legal tour. The new cycle crossover operator CX2 is compared with some traditional path representation methods such as order crossovers(OX) and partially mapped crossover (PMX). Various crossover operators have been presented for TSP with different applications by using GAs. The (PMX) and (OX) along with proposing crossover operator (CX2) are mainly focused on this article. Apply these three operators on a manual experiment and found that CX2 performs better than PMX and OX crossovers.

4-Sakkayaphop Pravesjit in 2017[10]. GA was Improved in this work for an optimization problem. A Gaussian function is used instead of the traditional operators in the crossover and the mutation. Five benchmarks used to test the algorithm. The results display that this algorithm is the best for all of them. It can solve the Multimodal function, the discontinuous function, and the continuous functions effectively. The computational results are compared with the other four algorithms: the traditional DE algorithm, the JDE self-adaptive algorithm, the self-adaptive differential evolution algorithm and the hybrid bat algorithm with natural-inspired algorithms. It shows that the proposed algorithm produces better results than the other four algorithms and provides the optimal solution for all tested functions.

5-Esra'a Alkafaween in 2018[11]. Presented a new mutation called "IRGIBNNM", that is hybrid of two mutations: a random-based mutation and a knowledge-based mutation. The proposed mutation used to improve Select Best Mutation (SBM) strategy. The proposed mutation performance has compared with three mutations, additionally to the SBM. To evaluate the performance of the proposed mutation and the SBM strategy, several experiments on twelve TSP instances are conducted. Those instances were taken from the TSPLIB. The experiential results show the efficiency of "IRGIBNNM" mutation, and the SBM strength, both methods benefit from the randomization and knowledge provided by the nearest neighbor approach.

6-Ahmad B. Hassanat in 2018[12]. The GA was improved by dividing large-scale TSP problem into small subsets based on regression. This is achieved by using the regression line and its perpendicular line, where cities are grouped into four sub-problems and frequently, each site locates the group in which the city belongs. This process is repeated until the size of the problem becomes very small or reaches a size that cannot be divided. To generate the initial population, This solution is mutated several times. Traditional population techniques, such as random and nearest neighbors, are used in addition to proposed, techniques for analyzing the performance of the GA. Some famous TSP instances that are Derived from TSPLIB are used in experiments. Quantitative analysis is performed using statistical test instruments: Duncan multiple range tests (DMRT), the least significant difference (LSD), and analysis of variance (ANOVA). The experiments show that the proposed method outperforms the other methods using population seeding techniques such as the nearest neighbor based techniques and random techniques regarding mean convergence and error rate.

7-Shuqu Qian in 2019[13]. A new Enhanced GA based on memory update systems and environmental reaction systems were proposed for solving constrained knapsack problems in dynamic environments (DKPs). To maintain a variety of memory solutions, when memory is updated, an elite that differs from existing memory solutions will supersede the worst solutions in total memory. In this case, the information can be stored as diverse as possible in total memory. Furthermore, the environment reaction process is used to learn how to use the saved solutions in the memory group and set recovery time. The experiment results are shown on a series of DKPs with various Dataset created at random. This proposed algorithm can track the changing environments faster and show distinct statistical performance.

3. Artificial Bee Colony (ABC)

ABC is a meta-heuristic optimization method a population-based. The population in the algorithm represent bees. Every bee searches for the best solution (food resource). Each solution is supposed as a food resource and the nectar amount of each resource represents the quality of each solution[14]. ABC is a population-based optimization algorithm that tries to achieve global minimum[15].

The ABC algorithm consists of two types of bees that are: employed bees and unemployed bees, which consist of onlooker bees, and the scout bees. The colony population usually consists of two parts, employed bees form the first half and the rest includes onlooker bees. Employed bees are responsible for exploiting the sources of nectar that has been explored by giving information to the onlooker bees in the hive about the quality of the sites of food sources they exploit. Based on the information shared by employed bees, the onlooker bees to decide on the source of the food to exploit it. Scouts study the environment randomly to locate a new food source based on potential external evidence or internal motivation.

The main algorithm of ABC are given below:

Begin

Step1: while termination condition not satisfied do
Generate initial population by Send the bees onto the food sources.
Evaluate the fitness f(x) of each solution x in population
Repeat
Step2 : Generate new solution by employed bees to the food sources and evaluate their nectar amounts
Step3: Place the onlooker bees on the food sources depending on their nectar amounts
Apply the greedy selection process for the onlookers
Step4 : Determine the abandoned solution for the scout, if exists,
Send the scouts for be discovered new food sources, randomly
Step5: Store the best food source found so far
Until the New population is complete
Replace the current population with the new population
End while
Step7: Return best solution
End

4.Genetic algorithms (GA)

Genetic algorithm (GA) is an optimization and search technique based on the concept of genetics and natural selection. At first, a genetic algorithm selects parents from an initial population of chromosomes to generate offspring using operations such as biological processes, crossover, and mutation. Evaluate all chromosomes using a fitness function to set their fitness. Then, fitness values are used to determine whether the chromosomes are discarded or keep. The less adaptive chromosomes are discarded and the most adaptive ones are kept when generating a new population according to the principle of survival of the fittest. The new population replaces the old. This process is repeated until a specific termination condition satisfied[16].

The main steps of the algorithm are given below:

INPUT: Initial population of n chromosomes
OUTPUT: Best solution
Begin
Step1: while termination condition not satisfied do
Evaluate the fitness f(x) of each chromosome x in population
Repeat
Select two parent chromosomes according to their fitness
Crossover the parents to form new offspring (children)
Mutate new offspring
Until the New population is complete
Replace the current population with the new population
End while
Step2: Return best solution
End

5. Statistical tests[17][18]

This part displays several tests intended to measure the goodness of a generator termed as a random bit generator. The operation is done by taking a sample from the output sequence and submitting it to several statistical tests. Each of the five tests determines whether a particular attribute sequence is likely to actually display a random sequence.

Let s = s0, s1, s2…sn−1 is a binary sequence of length n.

This paper presents five commonly used statistical tests to determine whether the s sequence has some specific features that are likely to show a real random sequence. The statistical tests that are used in this paper are:

1. The Frequency (Monobit) Test

For a random series of length n, the number 0's should be approximately equal to the number of 1's.

The statistic used in equation (1):

$$X1 = \frac{(s_0 - s_1)^2}{s} \quad \text{.....(1)}$$

Where: s0,s1 denote the numbers of 0's and 1's in s, respectively.

2. Serial (two-bit test) test

In this test, we determine if the frequency count is 00, 10, 01 and 11 are roughly the same.A Statistic used in equation (2):

$$X2 = \frac{4}{s-1}(s00^2 + s10^2 + s01^2 + s11^2) - \frac{2}{s}(s0^2 + s1^2) + 1 \quad \text{.....(2)}$$

Where s00, s10, s01 and s11 indicate the number of "00", "01", "10", and "11" in s, respectively.

3. Poker test

suppose p is a positive integer where [s /p] ≥ 5.(2p), and suppose D = [s /p]. Divide the sequence s into D non-overlapping parts each of length p, and suppose s_i is the counts of the appearance of the ith type of sequence of length p, 1 ≤ i ≤ 2p.. In this test, we determines if the length p sequence shows roughly the same number of times in s, as expected for the random sequence. The statistic used in equation (3):

$$X3 = \frac{2^p}{D}\left[\sum_{i-1}^{2^p} s_i^2\right] - D \quad \text{.....(3)}$$

4. Runs test

This test aims to determine if the number is either zeros or ones from a series of different lengths in sequence S as expected for a random sequence. The expected number of gaps (or blocks) of length i in a random sequence of length s is $e_i = (s - i + 3)/2i + 2$. suppose L be equal to the largest integer i for which ei ≥ 5. Suppose Bi, Gi are the number of blocks and gaps , respectively, of length i in n for each i, 1 ≤ i ≤ L.

The statistic calculated by equation (4):

$$X4 = \sum_{i=1}^{L} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{L} \frac{(G_i - e_i)^2}{e_i} \quad \text{.....(4)}$$

5. Autocorrelation test

In this test, we check the relationships between the sequence s and its shifted versions . suppose d is a fixed number , 1 ≤ d ≤ s/2. The number of bits in s not equal to their d-shifts is

$A(d) = \sum_{i=0}^{n-d-1} s_i + s_{i+d}$,

where ⊕ indicate the XOR operator .which roughly follows an N(0, 1) distribution if n - d ≥ 10. Since small values of A(d) are as unexpected as large values of A(d), we should be used a two-sided test . The statistic calculated by equation (5):

$$X5 = 2\left[A(d) - \frac{n-d}{2}\right]/\sqrt{n-d} \quad \text{.....(5)}$$

6. The proposed algorithm

This paper focuses on enhancing GA's initial population by using ABC to generate random keys. We will divide the work into two stages, the first stage will focus on generating random keys by using the ABC and the second stage will be generating random keys based on GA. We improved the previous hybridization work by generating an initial population by ABC then apply a crossover and mutation and select the best population from them as a new population for the next generation.

The five statistical tests are used to compute the fitness (each time any test is failed the counter of fitness will be incremented), such that the random keys with minimum fitness will be taken. GA is applied to the initial population. At last, the population produced from GA is compared with the initial population and selects the best population (replaces the original parental population) as the new population to the next generation. Reproduction imitates natural selection. The termination criterion of the proposed algorithm may reach the maximum cycle number or all the population is random numbers passed the tests. (See algorithm1).

6.1. ABC phase

This stage will generate numbers randomly within the boundaries of the parameters by using the following equation:

$$X_{ij} = X_j^{min} + rand(0,1)(X_j^{max} - X_j^{min}) , \qquad (6)$$

Where:

i = 1. . .SN , j = 1. . .D.

SN: is the solution number

D: is the number of optimization parameters

The created number is converted to binary. After this, the fitness function in the proposed algorithm is that the five statistical tests may be assigned to the Xij solution by Equation (7).

$$Fitness_i = \sum f_i \qquad (7)$$

Where fi is the counter value of the solution X_{ij}.

6.2. GA phase

GA performs genetic operators to generate offspring based on the initial population. The genetic process is performed Iteratively. The current population is sorted according to its fitness value, roulette wheel operation is used in the proposed algorithm. Crossover operator is an operator that combines two chromosomes (parents) to create a new chromosome (offspring). The conception of crossover is that a new chromosome possibly superior to both parents if it takes the best characteristics of both parents. Ordered crossover used in this algorithm. Mutation Operator is to change the genes of the offspring and to increase the diversity of the population. In the proposed algorithm we uniform mutation to generate a new gene (See algorithm2).

Algorithm1: *a proposed algorithm*

Input: key's length

Output: binary keys

Process:

Step1: Set the initial population using Eq(6) to be the current population. /* using ABC*/
Step2: Evaluate the current population using the five statistical tests to compute their fitness using Eq(7).

Step3: Repeat

Step4: Call algorithm 2 that takes the current population and produce new solutions using GA.

Step 5: Evaluate the new solutions that were generated from step 4 using the five statistical tests to compute their fitness using Eq(7).

Step6: Select the best populations between the solutions that generate from step4 and the current population to be a new population according to their fitness. /*Reproduction operation*/

Step7: Until the termination condition is met.

Step8: Return (a new population).

End.

Algorithm2: *GA phase algorithm*

Input: current population (binary numbers)

Output: new solutions

Process:

Step1: Sort the current population according to their fitness value. /* ascending order */

Step2: Select from the current population chromosomes by using the roulette wheel according to their fitness.

Step3: Perform crossover operation between the selected chromosomes generated from step2 to produce new solutions. /* ordered Crossover is used, with a crossover rate of 0.8*/

Step 4: If there is a mutation rate, perform mutation operation. /* uniform mutation is used, with a mutation rate of 0.01*/

Step 5: If a new solution is found in the current population repeat step 2 and step 4.

Step6: Return (new solutions).

End.

7.Results and discussion

In this paper, we enhance GA by using ABC for random key generation purpose. At first, we generate initial population by using ABC and its results are input to GA as current population also we tried using GA at first and its results are input to ABC as the current population, but we found that the hybrid between them as in the proposed method offers better results.

In the ABC stage, the number of employed bees is equal to the number of food sources. In GA, the rate of crossover is 0.8, and the mutation rate is 0.01. Computational experiments for all approaches were conducted on Intel core i-3 machine and coded in python3.

We test our proposed method on data consist of 64, 120, 128, 192, and 256 key's length respectively. The process is executed 5 times with various recommended parameters for the three variations of data, we set the population size 10, 20, 30, respectively, and the maximum iteration number (MIN) is 100.

The goal is that all population is random according to the five statistical tests. It is based on the execution time and the minimum iterations to achieve the best solution (see eq. 9).

Mean=($\sum_0^n$ rate)/n ----------(9)

Where rate is the number of iterations to achieve stop condition, and n is the number of run times.

Tables 1, 2 and 3 presented the summary of the results. The first column points to the size of the keys. In the second and fourth columns, the average number of iterations of 5 attempts for the traditional GA and proposed algorithm respectively. The third and fifth columns indicate the average time taken to implement the traditional GA and proposed algorithm respectively. And the sixth column indicates the difference between the mean iteration of the traditional genetic algorithm and the proposed algorithm.

The results show that this enhanced method is relatively better than its original method with traditional GA.

Table1: Comparison of traditional GA and enhanced GA (population size = 10)

key size	traditional GA		proposed algorithm		Difference
	Mean iteration	Time	Mean iteration	Time	
64	5.6	0.0361	2.6	0.0186	3
120	14.8	0.1716	7.2	0.0515	7.6
128	15.4	0.1058	6.8	0.0594	8.6
192	24.4	0.3142	15.8	0.1466	8.6
256	30.4	0.4808	20	0.2392	10.4
mean	18.12	0.2217	10.48	0.10306	7.64

Table2: Comparison of traditional GA and enhanced GA (population size = 20)

key size	traditional GA		proposed algorithm		Difference
	Mean iteration	Time	Mean iteration	Time	
64	4.4	0.037	3	0.031	1.4
120	12.6	0.1886	6.8	0.122	5.8
128	15	0.2366	7.4	0.1234	7.6
192	29.8	0.6984	17.8	0.4334	12
256	30.8	0.9542	17.6	0.6116	13.2
mean	18.52	0.42296	10.52	0.26428	8

Table3: Comparison of traditional GA and enhanced GA (population size = 30)

key size	traditional GA		proposed algorithm		Difference
	Mean iteration	Time	Mean iteration	Time	
64	5.2	0.0666	2.2	0.039	3
120	11.6	0.243	6.6	0.1598	5
128	14.8	0.352	7.4	0.196	7.4

192	24	0.83	15	0.5424	9
256	30.6	1.5752	20	1.048	8.6
mean	17.24	0.61336	10.24	0.39704	6.6

8. conclusion

The following points can be concluded:

1-Using a hybrid between GA and ABC will enhance GA in generating random keys.

2-Using the five statistical tests provides a good fitness function for random keys generating purpose .

3-The large size of a population is appropriate for the proposed algorithm, as the results show, the difference between traditional GA and proposed algorithm increase when key size increase and decrease when the population size is decreasing.

4-The final keys that are generated are unique (not repeated), random and cryptographically strong (successfully passed the five statistical tests).

5-Use ABC to generate the initial population for GA, and select the best solutions from GA and ABC to be the population of the next generation, which will increase the efficiency of generating random keys through minimizing the number of iteration that is required.

References

[1] Fadheela Sabri Abu-Almash, "Apply GA for Pseudo RNG_2016.pdf." International Journal of Advanced Research in Computer Science and Software Engineering, pp. 8–19, 2016.

[2] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer …, 2005.

[3] C. Liu and L. H. Fan, "Conformal Array Pattern Synthesis Using a Hybrid GA/ABC Algorithm," in *2015 International Conference on Artificial Intelligence and Industrial Engineering*, 2015.

[4] S. Jawaid, A. Saiyeda, and N. Suroor, "Selection of Fittest Key Using Genetic Algorithm and Autocorrelation in Cryptography," *J. Comput. Sci. Appl.*, vol. 3, pp. 46–51, 2015.

[5] J. S. Geetha and D. I. G. Amalarethinam, "ABCRNG-Swarm Intelligence in Public key Cryptography for Random Number Generation," *Int. J. Fuzzy Math. Arch.*, vol. 6, no. 2, pp. 177–186, 2015.

[6] M. Alkharji, M. Al Hammoshi, C. Hu, and H. Liu, "Genetic Algorithm based key Generation for Fully Homomorphic Encryption," in *Proceedings of 16th Annual Security Conference*, 2017.

[7] Y. Deng, Y. Liu, and D. Zhou, "An improved genetic algorithm with initial population strategy for symmetric TSP," *Math. Probl. Eng.*, vol. 2015, 2015.

[8] A. Shrestha and A. Mahmood, "Improving genetic algorithm with fine-tuned crossover and scaled architecture," *J. Math.*, vol. 2016, 2016.

[9] A. Hussain, Y. S. Muhammad, M. Nauman Sajid, I. Hussain, A. Mohamd Shoukry, and S. Gani, "Genetic algorithm for traveling salesman problem with modified cycle crossover operator," *Comput. Intell. Neurosci.*, vol. 2017, 2017.

[10] S. Pravesjit and K. Kantawong, "An improvement of genetic algorithm for optimization problem," in *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)*, 2017, pp. 226–229.

[11] E. Alkafaween and A. B. A. Hassanat, "Improving TSP Solutions Using GA with a New Hybrid

Mutation Based on Knowledge and Randomness." pp. 1–18, 2018.

[12] A. Hassanat, V. Prasath, M. Abbadi, S. Abu-Qdari, and H. Faris, "An improved genetic algorithm with a new initialization mechanism based on regression techniques," *Information*, vol. 9, no. 7, p. 167, 2018.

[13] S. Qian, Y. Liu, Y. Ye, and G. Xu, "An enhanced genetic algorithm for constrained knapsack problems in dynamic environments," *Nat. Comput.*, pp. 1–20, 2019.

[14] T. Dokeroglu, E. Sevinc, and A. Cosar, "Artificial bee colony optimization for the quadratic assignment problem," *Appl. Soft Comput.*, vol. 76, pp. 595–606, 2019.

[15] T. Rashid and S. Abdullah, "A Hybrid of Artificial Bee Colony, Genetic Algorithm, and Neural Network for Diabetic Mellitus Diagnosing," *ARO-The Sci. J. Koya Univ.*, vol. 6, no. 1, pp. 55–64, 2018.

[16] S. N. Sivanandam and S. N. Deepa, "Genetic algorithms," in *Introduction to genetic algorithms*, Springer, 2008, pp. 15–37.

[17] A. J. Menezes, "PC van," *Oorschot, SA Vanstone, Handb. Appl. Cryptogr. CRC*, 1996.

[18] G. Carter, "Statistical tests for randomness," in *Proccedings of the Workshop on Stream Ciphers. Report*, 1989, vol. 89, no. 1.