مجلة

كلية التـــراث الجامعة

مجلة علمية محكمة

متعددة التخصصات نصف سنوية

العدد الثامن والثلاثون

15 شباط 2024

ISSN 2074-5621

رقم الايداع في دار الكتب والوثائق 719 لسنة 2011

مجلة كلية التراث الجامعة معترف بها من قبل وزارة التعليم العالي والبحث العلمي بكتابها المرقم (ب 3059/4) والمؤرخ في (4/7 /2014)

Efficient Latin Square Construction via Evolutionary Algorithms

Fouad Sahib Muhamed

Ministry of Higher Education and Scientific Research

Abstract— Evolutionary or genetic algorithm mimics nature's principles of evolution to find the optimal solution to any problem that is compatible with its mechanism. The genetic algorithm (GA) is one of the most well-known heuristic methods for resolving computing problems. We discover that Darwinian evolution's characteristics have been expressed after studying the GA. It has actually had a number of achievements in various areas of life's demands. The magic square (MS), which is an array of positive integers (1, 2, n) configured so that the sum of the n numbers in any principal diagonal, horizontal, or vertical line is always the same number, was correctly obtained for this study using a GA. In particular, evolutionary algorithms have been utilized as a way to find the best MS solution. How successfully does GA optimization accomplish the MS is the research subject. This study has successfully established a way to build Latin Squares that can adapt to variable sizes (N x N), allowing them to handle Latin Squares of various lengths. This novel method offers a more adaptable method for dealing with Latin Squares of any scale, diverging from the traditional emphasis on specific numbers or defined dimensions.

Keywords— MS; GA; fitness function; evolution; Optimization.

1. Introduction

The GA that was built to find the MS has produced positive results. GA has imitated Darwin's theory of natural selection and evolution, which holds that life evolves slowly and gradually from non-life or simple life (simple solution in GA) into optimal life (optimal solution in GA). It will be quite difficult to describe problems where the GA and construction are not properly suited. However, the design of the MS is extremely apparent, and a GA might potentially improve it. In order to acquire a wide variety of solutions, crossover operation has been used. The algorithm has been implemented using the structure of vertical and horizontal arrays.

2. Darwin's Theory

GA has replicated two key elements of Darwin's theory: natural selection and the evolutionary process. According to Darwin's Theory of Evolution, the origin of life was either non-life or simple life (a simple solution in GA), and the growth of life has been a slow, gradual process (optimal solution in GA). In other words, complicated makes naturally developed over time from more straightforward lines (Darwin, 1859). Minor advantageous genetic mutations are kept and gathered by natural selection. Let's say that a member of classes developed a trait. For instance, it developed wings and acquired flight skills. Its children would inherit those traits, which would then be passed down to their children, and so on. The inferior (traits) members of a class would gradually disappear, and only the superior (traits) members of the class would survive. Natural selection refers to the preservation of traits that enable a class to compete more successfully in the wild. It more closely resembles domestic breeding. Human breeders have dramatically altered domestic animal populations throughout history by selecting particular species to breed. Breeders gradually eliminate unfavorable characteristics over time. Additionally, over time, natural selection gradually eliminates the inferior class. Here is an example that will provide further information. There is a population of rabbits in the wild, some of which are intelligent, some of which are not, and some of which are swift, some of which are not. Foxes are more likely to eat the slower, dumber rabbits. The quick and intelligent ones, however, have a better chance of surviving and reproducing to produce a new generation of rabbits. Of course, some of the less intelligent and slower rabbits may survive, perhaps by chance, but their numbers will be lower than those of the quick and intelligent ones. Natural selection, of which foxes are a part, is what leads to the smart and quick rabbits being much more prevalent in the wild than other sorts of animals over time since there are more parents of their type (Michalewicz, 1996) (Michalewicz & Fogel, 2004).

3. Optimization

Finding the most advantageous parameters for a given model is the goal of the problem-solving technique known as optimization. The optimizer is aware of the model, which takes inputs and produces outputs. Typically, the problem can be framed so that we aim to minimize the model's output value or the output of a function that converts the model's output into a fitness score. Because of this, the procedure is frequently referred to as minimization. It becomes clear that this is helpful when thinking about how to optimize a circuit's design in order to reduce power usage. To do this, the optimizer searches for parameter combinations that allow the model to provide the optimum output given a given input. Analytical techniques can be used to optimize when dealing with simple mathematical models, frequently by computing the derivative of the functional model. However, these techniques are challenging to apply to complicated models that have noisy behavior. Additionally, it is impossible to employ such procedures because the analytical model is not always known. Algorithms that are effective at resolving these types of optimization issues can be found in the discipline of evolutionary computation (EC), which is a subset of computational intelligence (CI), which is in turn a subset of artificial intelligence (AI) (Dahlberg, 2017).

4. Optimization Algorithms Classifications

There is a need for robust, adequate, and global algorithms to solve optimization problems such as computer science and engineering problems. To deal with these problems, there are two classifications of algorithms: stochastic and deterministic algorithms. The first one usually provides approximate solutions but is not optimal. The latter, probabilistic algorithms are often recommended to be used probabilistic algorithms. It does not give for sure the optimal solution. In fact, it generates a randomly highly accurate solution with better performance (Agrawal et al., 2022).

5. Evolutionary Algorithm (EA)

The focus of evolutionary computation is on natural process-inspired problem-solving algorithms. The fundamental tenet of the area is to apply biological Darwinian evolution's mathematical models to optimization issues. Imagine that an organism functions as a "input" to the "model" of its natural environment and produces a "output" in the form of offspring to demonstrate how useful this is. Through repeated iterations, biological evolution purges the population of organisms, preserving only the fit individual, to develop species that steadily improve their environmental adaptation. However, evolutionary computation is not just limited to Darwinian evolution; it also covers a wide range of techniques that draw inspiration from other natural phenomena like cultural evolution and animal behavior. Populations are the basis of evolutionary algorithms. In optimization issues, a population is a group of individuals that has a vector of parameters that the model we want to optimize can accept and use to produce an output. A process is used to initialize the population with a random set of parameter vectors that should uniformly cover the model's whole parameter range. Once the first population has been assessed, an iterative process is initiated and continues until a workable solution is discovered. This cyclical process involves choosing, changing, and assessing the current population. During selection, a group of individuals who exhibit promising traits are chosen to survive into the population's next generation. They are then examined after being randomly altered to add diversity to the population. Each iteration of this process results in the creation of a new generation of the population, and it continues indefinitely until a solution is discovered or another restriction is met (Dahlberg, 2017).

6. Genetic Algorithm (GA)

One of the most popular optimization metaheuristics methods to solve computational problems is the genetic algorithm (GA). It mimics Charles Darwin’s theory of evolution that gets controlled by natural selection. It works by the principle that the fittest or the one which is so compatible with the natural conditions is the most one to get survived. It is a stochastic and adaptive search method that has been developed by John Holland. Numerous challenging numerical optimization problems have been successfully solved using GA. Problems with system identification, signal processing, and path planning have all been effectively solved using it. Comparing GA to traditional search algorithms, one of the main benefits of using it is that they operate on a population of solutions rather than just a single point. As a result, GA results are more reliable and precise. The answer offered by GA is more ideal and comprehensive in scope. Local optima, such as those produced by Newton or gradient descent algorithms, are less likely to trap GA. No derivative fitness criterion information is needed for GA. This makes it a great fit for situations involving both continuous and discrete optimization. Furthermore, GA is less sensitive to measurement noise and ambiguity (Agrawal et al., 2022).

6.1 GA Steps:

1- Generate initial population.
2- Fitness evaluation.
3- Selection.
4- Crossover and Mutation.
5- Formation of a new population.
6- Termination (Agrawal et al., 2022).

GA works by initiating generations and each generation contains a number of chromosomes (individuals). Each individual contains genes that represent a solution to a problem. The process of evolution is applied to each individual to get new offspring and that depends on choosing the good one or the one that fit the solutions after combining them. The process of combining and choosing fit chromosomes continues to find the optimal solution which is the optimal chromosome. A population of individuals is used in the GA search process, and each one is assessed according to its fitness value. Higher-fit individuals are chosen to have offspring that have many, but not all, of their parents' characteristics. Utilizing genetic operators such as crossover and mutation to accomplish this search process. Genetic search algorithms differ from traditional search algorithms in a few key ways:

1- GA carries out the search utilizing a coded solution rather than the actual solutions.
2- GA evaluates individuals based on their fitness function rather than the derivative of the function and are based on a population of possible solutions rather than just one.
3- GA employs probabilistic operators rather than deterministic ones, such as crossover and mutation (Agrawal et al., 2022).

6.2 Process of Evolution:

The fitness of each member of the initial population is calculated to begin the evolutionary process of GA. The following actions are taking while the stopping requirement is still not yet met:

1- Select an individual utilizing some selection processes for reproduction (i.e. tournament, rank, etc.).
2- Use the crossover and mutation operations to produce a progeny.

Based on the application, the crossover and mutation probabilities are chosen. The newest generation, compute. This process will come to an end in case either the best answer is discovered, or the maximum number of generations has been achieved (Agrawal et al., 2022).

7. Magic Square (MS)

An unidentified mathematician invented the MS originally in China. The first MS of order three was called Lo Shu. It tells the tale of a flood that occurred in China before three thousand years ago. At that time, people attempted to make an offering to the enraged river deity, but he did not respond. Every time they offered a suggestion, a turtle would emerge from the riverbank. One day a boy noticed a mark on the turtle's back that represented the numbers 1 through 9. The numerical quantities were arranged in a line, with 15 added to each square of numbers. People at the moment discovered that the amount they had given wasn't the proper one (Swaney, n.d.).

7.1 MS Kinds

It can be classified depending on construction way into three types:

1- Additive MS: it is the standard one and its elements are arranged in a way when we find the sum of the columns, rows, and diagonals gives equal sums (S) as shown in Table 1.
2- Multiplicative MS: it contains elements that get arranged along columns, rows, and diagonals to multiply to get a magic product called (p) as shown in Table 2.
3- Additive-multiplicative MS: it contains elements respectively arranged along columns, rows, and diagonals to get added and multiplied for obtaining (S) and (P) as shown in Table 3 (Johnson, 2005).

Table (1): 5 × 5 additive MS with magic sum 65

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Table (2): 3 × 3 multiplicative

MS with p 4096

128	1	32
4	16	64
8	256	2

Table (3): 8 × 8 additive multiplicative MS : S (840), P (2,058,068,231,856,000).

64	83	117	102	15	76	200	203
19	60	232	175	54	69	153	78
216	1617	17	52	171	90	58	75
135	114	50	87	184	189	13	6
150	261	45	38	91	136	92	27
119	104	108	23	174	225	57	30
116	25	133	120	51	26	162	203
39	34	138	243	100	29	105	152

It can be classified depending on the elements involved into two types:

1- Non-repetitive MS: it contains elements that are different from each other and not repeated. it is the same as standard MS and the number of its elements is the number of columns multiplied by the number of rows as shown in Table 4.
2- Repetitive MS: it contains elements that have to be used in all columns, rows, and diagonals, one time in each one for all of them without getting repeated in all columns, rows, and diagonals. in this type of MS, the number of elements is equal to the number of rows and equal to the number of columns. also, the number of elements is less than the number of rows multiplied by the number of columns as shown in Table 5 (Johnson, 2005).

Table (4): 3 × 3 non-repetitive MS with magic sum 15.

8	1	6
3	5	7
4	9	2

Table (5): 5 × 5 repetitive MS with magic sum 30.

9	6	3	5	7
6	7	5	9	3
5	3	6	7	9
7	5	9	3	6
3	9	7	6	5

It can be classified depending on its order into two types:

1- Even order MS: it has equal rows and columns and its order of n is equal to 2k with k being bigger or equal to number 2 as shown in Table 6.
2- Odd order MS: it has equal rows and columns and its order of n is equal to 3k with k being bigger or equal to number 1 as shown in figs 1, 2, 4, and 5 (Johnson, 2005).

Table (6): 4 × 4 even-order MS with magic sum 34.

1	14	11	8
12	7	2	13
6	9	16	3
15	4	5	10

We can say that figures 1, 2, 4, and 5 are related to an odd order, whereas figures 3 and 6 are of an even order. figure 3 is a non-repetitive of even order and additive-multiplicative MS (Johnson, 2005).

8. Related Work

(Johnson, 2005) describes how Latin squares are used to construct trials with interacting elements, providing balanced and effective designs. It also investigates its use in statistical modeling, notably in network-based complicated data structures, and offers a helpful paradigm for modeling dependent data. Networks are structures that in the context of the paper reflect connections or interactions between elements. The study demonstrates the value of Latin squares in network analysis and experiment design. Their adaptability and strong influence in numerous statistical applications are highlighted. The work also investigates relationships between Latin squares and other branches of mathematics, like graph theory and block designs, highlighting their contributions to progress in these branches. Latin squares are used in network analysis and statistical investigations, and the author uses real-world examples and case studies to illustrate the principles being discussed. The research concludes by summarizing its major findings and emphasizing the value of Latin squares in solving challenging network-related issues. Overall, "The Role of Latin Squares in the Study of Networks" is an invaluable tool for statisticians and academics working on network and experiment design analyses.

(Yan, 2005) describes a novel approach for creating typical magic squares. The rows, columns, and diagonals of these magic squares are all consecutive integers beginning with 1, and they add up to a fixed value. The authors' goal is to propose an original and effective technique for producing normal magic squares of various orders. The new algorithm is thoroughly explained in the paper, with each stage of its process being broken down and its advantages and disadvantages over existing approaches highlighted. The authors explore the fundamental ideas and characteristics of regular magic squares, which the method uses, and then provide mathematical analyses and proofs to demonstrate the correctness and efficiency of their technique. The study also presents actual data and numerical examples showing the performance of the algorithm in generating normal magic squares of various sizes. The authors demonstrate the benefits and distinctive characteristics of their novel algorithm by contrasting it with existing widely used techniques for creating typical magic squares. They underline how effective a tool it may be for producing typical magic squares. The paper discusses the significance of normal magic squares in recreational mathematics, number theory, and coding theory, along with possible extensions to other areas. The development of this novel algorithm, which provides a creative and effective solution to a traditional mathematical problem, is the paper's main contribution. The research concludes by summarizing its key findings and stressing the benefits of the suggested approach for creating normal magic squares of varied sizes. It emphasizes the algorithm's worth for mathematicians and math fans interested in magic squares and related disciplines. Overall, "A Novel Algorithm for Constructing Normal Magic

Squares" gives a novel and efficient method for creating normal magic squares, making it an important tool for researchers and mathematicians in the investigation of this fascinating mathematical subject.
(Melas & Karamanos, 2009) intention is to provide a succinct and narrowly focused remark about diagonally magic squares, emphasizing their unique qualities and properties. An example of a diagonally magic square is one in which the sums of the numbers in each row, column, and major diagonal are equal, as well as the sums of the numbers in the secondary diagonals. This essay examines diagonally magic squares mathematically, explaining how they are made and why they exist. It examines the requirements and limitations that must be met to create a diagonally magic square. To further clarify the idea and show how they differ from conventional magic squares in terms of intriguing qualities, the author also provides properties and instances of diagonally magic squares. The importance of diagonally magic squares is highlighted, along with some of the prospective fields in which they could be used, especially in the realm of mathematical puzzles and problem-solving. By emphasizing this less well-known variation and throwing light on this distinct subclass of magic squares, the work adds to the body of literature on magic squares and advances our knowledge of their mathematical features. The paper outlines its main findings and deductions on diagonally magic squares in its conclusion. It highlights how crucial more study of this particular kind of magic square is, as well as any potential mathematical ramifications. Overall, "Note on a Diagonally Magic Square" contains insightful information about this special type of magic squares and is likely to be of interest to mathematicians, researchers, and those who are just fascinated by the intriguing qualities of magic squares.
(Wu & Wu , 2019) explore two remarkable issues in recreational mathematics, knight's tours and magic squares, and their surprising connection. Knight's tours are a series of moves made by a knight on a chessboard, visiting each square precisely once, whereas magic squares are collections of numbers arranged so that the sums of the numbers in each row, column, and diagonal are equal. The writers' goal is to explore the relationship between knight's tours and magic squares, providing fresh perspectives on these related ideas. The paper starts with an overview of knight's tours and discusses alternative construction methods for them on various chessboard sizes, including time-honored techniques like Warnsdorff's rule and heuristic algorithms. The intriguing relationship between knight's tours and magic squares is explored in this study, showing how some knight's tours can be used to create magic squares and vice versa. This investigation demonstrates the intricate mathematical connections between these two seemingly unconnected subjects. The relationship between knight's tours and magic squares is supported by mathematical analysis and arguments, which reveals special and intriguing characteristics in the generated magic squares. The paper examines potential generalizations of the study's insights together with its theoretical conclusions and real-world implications. It demonstrates how the lessons learned from this study can be applied to different chess piece tours and combinatorial designs. Knight's tours and magic squares are now better understood thanks to this research's discoveries of intriguing links that add to our understanding of recreational mathematics. The study concludes by summarizing the main conclusions and learnings from the analysis, emphasizing the grace and beauty of these mathematical constructions, and urging future research in this fascinating area. Overall, the study reveals fascinating parallels between knight's tours and magic squares while also being interesting and

instructive. For math lovers and others curious to learn more about the world of amusing puzzles, it makes for a pleasant read.
(Melas and Karamanos , 2009) the Latin and Graeco-Latin square concepts, specific varieties of combinatorial patterns, are used in the study to propose a modern way for creating magic squares. The goal of this work is to use Latin and Graeco-Latin squares to offer a novel method for creating magic squares. In combinatorial mathematics, experimental design, cryptography, and coding theory, Latin squares are collections of symbols or numbers that are arranged so that each symbol occurs precisely once in each row and column. By employing two sets of symbols, Graeco-Latin squares expand on this idea by making sure that each pair of symbols from the two sets appears precisely once in each row and column. The authors suggest a method that methodically converts Latin and Graeco-Latin squares into magic squares of various orders. They show the fascinating linkages between combinatorial designs and magic square creation and give a complete algorithm and step-by-step instructions for building magic squares using this method. The paper's main contributions are in providing a new viewpoint on the construction of magic squares by demonstrating how Latin and Graeco-Latin squares might be used. For scholars and math lovers who are interested in these mathematical constructions, the proposed method broadens their toolkit. The paper's conclusions have consequences for a number of disciplines, including magic squares' applications in coding theory, cryptography, and recreational mathematics. To create magic squares of various sizes, combinatorial designs offer a strong and effective method. The study concludes by listing the benefits of the modern way to making magic squares over more conventional approaches. The technique gives a more organized and methodical way to make magic squares by integrating Latin and Graeco-Latin squares, adding to the body of knowledge already known about this age-old mathematical conundrum. Overall, the study makes an important and original contribution to the subject of magic squares by offering new perspectives on how they are made and highlighting the exciting connections they have with combinatorial designs.
In conclusion, the collection of works addressed in the connected research demonstrates the numerous uses and intriguing characteristics of magic squares. Latin squares are essential for creating balanced trials and investigating network-based data structures, as Johnson's study emphasizes. An efficient method for producing regular magic squares is introduced by Sun and Feng's creative algorithm , while Yan's investigation clarifies the unique characteristics of diagonally magic squares. The unexpected links between knight's tours and magic squares are explored in Wu and Wu's work . By utilizing Latin and Graeco-Latin squares to build magic squares of various orders, Melas and Karamanos' unique methodology offers a novel viewpoint on their construction. Collectively, these research promote experiment design, network analysis, and recreational mathematics, furthering the development of mathematical understanding and its practical applications.

9. Research Methodology

It will be quite challenging to tackle problems whose structures do not fit with those of the GA. However, the classical MS structures are so distinct that using evolutionary algorithms to optimize will be a very practical approach. The main aim of this paper is to create method to obtain classical MS with order n x n by utilizing the steps of evolutionary algorithm (GA). The method depends on stochastic and diversity to enlarge the solutions space. It uses heuristic thought and checks the progress periodically to drive the process toward the optimal solution.

The method receive different n order depending on the user demand. It provides the opportunity to the program to think and that by trying, checking, and decide to how to pick the current choice or uses another choice. The structure of classical MS is so convenient to be evolved by GA. Each row and column could be represented as an chromosome in GA. The swap between its element represents the crossover and the swap could be between rows, columns, or cells itself.

The programing language has been used to solve the problem of MS by GA is java language. Many arrays have been used to apply the algorithm. Number of methods have been used to carry out the program such as:

1- enterMSvalues: method is used to initialize the MS members and it receive the order of MS as a parameter.
2- writeResults: method is used to print the result of GA on the screen and it print the results in each time of GA steps. It useful method that it shows the progress of the program whether it going up or down toward the optimal solution.
3- crossover: method to interchange the values between the arrows and columns, which is a Single Point, the crossover method has been employed. The way crossover works is to pick a random position for the columns or arrows, then swap all the rooms in that direction between two of them. To broaden the range of potential solutions and obtain additional diversity, a random procedure is used.
4- swap: method was employed to exchange the values of the arrows with the columns after the initial matrix was completed, with all the arrows organized such that they all have the same summation for their values.
5- obtainDiagonal: The desired summation for the matrix's diagonal has been obtained using this method.

10. Results and Discussion

Several classes from the Java library were imported and used early on in the application. In the Java library, these classes were arranged into packages to ensure a well-structured approach. The procedure for creating the final magic square got under way with the completion of GA stages. Following the rules of object-oriented programming, method strategies were used throughout the program to make the stages of the algorithm more understandable and manageable.

Repeated calls to each method were made during the execution of the program in an effort to gradually converge to the ideal magic square. In particular, the crossover method's evaluation of the fitness value resulted in a set of instructions connected to the fitness function of the GA. These instructions evaluated whether the algorithm was heading in the right direction or veering off course dynamically. These rules guided the program's development as it worked to find the optimum answer. This key mechanism served as the program's decision-making hub, selecting whether to use the same crossover technique or a different random crossover strategy. This component could be viewed as the program's cognitive hub.

During execution, the program changed from a state where decisions were made arbitrarily to one where it was getting closer and closer to the perfect decision. Incremental results were produced as the algorithm iteratively improved its solution. The program prints the magic square both before and after performing the genetic algorithm, as shown in the visual examples below:

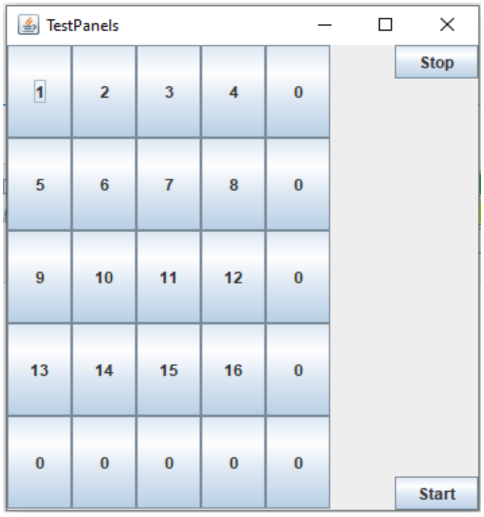

Figure 1– shows MS with order 4 x 4 after initiates its values.

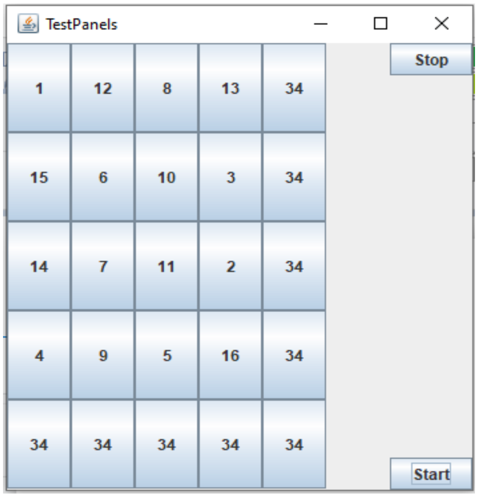

Figure 2 – shows MS with order 4 x 4 after applying GA.

Here is another example printed:

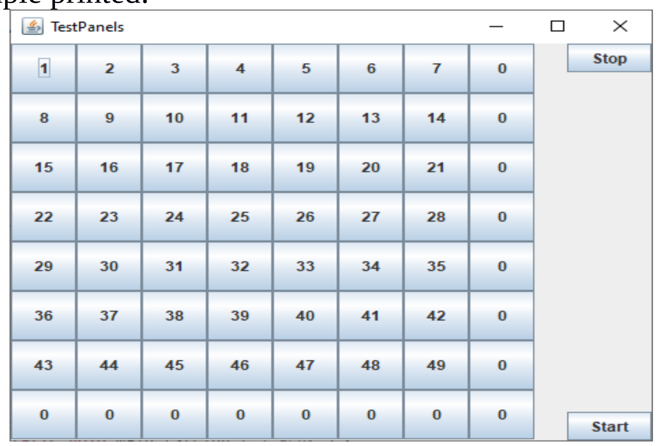

Figure 3– shows MS with order 7 x 7 after initiates its values.

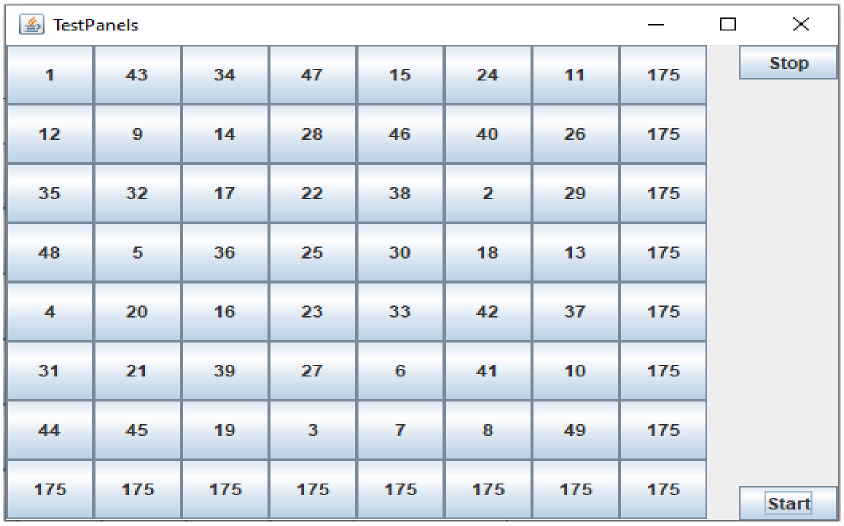

Figure 4– shows MS with order 7 x 7 after applying GA.

11. Conclusion

The Darwinian theory's mechanisms of evolution and natural selection were examined in this study. The evolutionary algorithms, a techniques for locating the best answer to some problems in the actual world where these problems have a structure suitable to them, was explored in relation to how to create the magic square. To carry out its processes, it needs stochastic and variety. It uses the crossover operation to get a variety of solutions, and it uses the fitness function as the algorithm's brain to manage everything. The study software was created using the Java programming language. The results demonstrated that the genetic algorithm is a successful approach to carrying out and obtaining the magic square, which provided a solution to the research topic.

12. References

1. Darwin, C. (1859). On the Origin of Species by Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. London: John Murray.
2. Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs (3rd ed.). Springer-Verlag Berlin Heidelberg.
3. Michalewicz, Z., & Fogel, D. (2004). How to Solve It: Modern Heuristics (2nd ed.). Springer-Verlag Berlin Heidelberg.
4. Dahlberg, L. (2017). Evolutionary Computation in Continuous Optimization and Machine Learning (Student's thesis). Mälardalen University, School of Innovation Design and Engineering.
5. Agrawal, N. K., Yadav, S., Gupta, A. A., & Pandey, S. (2022). A genetic algorithm model for optimizing vehicle routing problems with perishable products under time-window and quality requirements. Decision Analytics Journal. https://doi.org/10.1016/j.dajour.2022.100139
6. Swaney, M. (n.d.). Mark Swaney on the history of magic squares [Online]. Available from http://web.archive.org/web/20040807015853/http://www.netmastersinc.com/secrets/magic_squares.htm [Accessed 11th August 2017].
7. Johnson, N. L. (2005). The Role of Latin Squares in the Study of Networks. The Annals of Statistics, 33(3), 973-1006.

8. Yan, J. (2005). Note on a Diagonally Magic Square. Applied Mathematics Letters, 18(8), 941-944.
9. Melas, V. B., & Karamanos, K. D. (2009). A Modern Approach to Magic Squares Using Latin and Graeco-Latin Squares. Journal of Statistical Planning and Inference, 139(3), 1180-1191.
10. Wu, C., & Wu, T. (2019) explore Knight's tours and magic squares in their study published in Mathematics Magazine.
11. Melas and Karamanos (2009) present a modern approach to magic squares using Latin and Graeco-Latin squares in the Journal of Statistical Planning and Inference.

الحصول على المربع اللاتيني أو المربع السحري باستخدام الخوارزمية التطورية

فؤاد صاحب محمد
وزارة التعليم العالي والبحث العلمي , بغداد, العراق

المستخلص

تحاكي الخوارزمية الجينية او التطورية مبادئ التطور في الطبيعة لا يجاد الحل الامثل لأي مشكلة تتوافق ما اليتها. حيث تعد الخوارزمية الجينية واحدة من اكثر الطرق الاستكشافية شهرة لحل مشاكل الحوسبة. وقد لوحظ ان خواص نظرية دارون قد تم تفعيلها بعد دراسة الخوارزمية الجينية. قد حققت الخوارزمية الجينية العديد من الانجازات في مختلف متطلبات الحياة. المربع السحري هو عبارة عن مصفوفة من الارقام الموجبة (1, 2, ..., ن) بحيث ان مجموع اي صف سواء كان عمودي , افقي , او قطري هو نفس الرقم دائما وقد تم في هذا البحث الحصول عليه بواسطة الخوارزمية الجينية. بصورة عامة في هذا البحث قد تم استخدام الخوارزمية التطورية للحصول على المربع السحري وان محور البحث هو هل ان الخوارزمية الجينية او التطورية مناسبة للحصول على المربع السحري. قد نجحت هذه الدراسة في إنشاء طريقة لبناء مربعات لاتينية يمكنها التكيف مع الأحجام المتغيرة ، مما يسمح لها بالتعامل مع المربعات اللاتينية ذات الأطوال المختلفة. تقدم هذه الطريقة الجديدة طريقة أكثر قابلية للتكيف للتعامل مع المربعات اللاتينية من أي مقياس ، متباينة عن التركيز التقليدي على أرقام محددة أو أبعاد محددة.