ANALYSIS OF THE EVOLUTION OF GENETIC ALGORITHMS

Jayshri Kushwah
Research Scholar
Department of Mathematics
Kalinga University, Naya Raipur.
jayshrikush01@gmail.com

Dr. Rahul Dwivedi
Associate Professor
Department of Mathematics
Kalinga University, Naya Raipur.

Abstract

This article examines the most recent advancements in genetic algorithms. The most intriguing genetic algorithms have been selected by scientists for study. The capacity of this review to provide a larger viewpoint on genetic algorithms will be advantageous to the young and demanding researchers. The use of well-known algorithms is shown, along with each algorithm's advantages and disadvantages. The genetic operators and their applications are provided to assist new researchers. The numerous research areas connected to genetic algorithms are covered. The opportunities for research on hybrid algorithms, fitness measurements, and genetic operators are underlined. The graduate teaching and research communities will benefit from this well-organized review.

***Keywords**: Optimization, Metaheuristic, Genetic algorithm, Crossover*

Introduction

In recent years, real-world difficult issues emerging from several domains, such as engineering, economics, politics, management, and engineering, have been solved using metaheuristic algorithms. The main components of a metaheuristic algorithm are intensification and diversity. To effectively tackle the real-life situation, a balance between these factors is necessary.

The majority of metaheuristic algorithms get their inspiration from physics law, swarm behavior, and biological evolution. These algorithms may be generally divided into two groups: population-based metaheuristic algorithms and single solution algorithms. Single candidate solutions are used by single-solution based metaheuristic algorithms, and local search is used to enhance these solutions.

The result of single-solution based metaheuristics, however, may persist in local optimum. These metaheuristics prevent solutions from being trapped in local optima and preserve population variety. The genetic algorithm, particle swarm optimization, ant colony optimization, spotted hyena optimizer, emperor penguin optimizer, and seagull optimizer are a few examples of well-known population-based metaheuristic algorithms.

One well-known metaheuristic algorithm that draws its inspiration from biological evolution is the genetic algorithm. Genetic Algorithm apes Darwin's principle of the "survival of the fittest" in the natural world. J.H. Holland suggested Genetic Algorithm in 1992. Chromosome representation, fitness selection, and biologically inspired operators make up the fundamental components of Genetic Algorithm. Inversion, a unique component that is often employed in Genetic Algorithm implementations, was also presented by Holland. The chromosomes often take the shape of a binary string. Each locus on chromosomes has two potential alleles: 0 and 1. Considered as points in the solution space are chromosomes. These are handled by the repeated replacement of its population employing genetic operators. Each chromosome in the population is given a value using the fitness function.

Research methodology

The review of Genetic Algorithm was conducted according to PRISMA's recommendations. To find academic publications on genetic algorithms, a thorough search has been conducted on Google Scholar and PubMed. This document also includes significant research articles that were discovered via manual search. Keywords like "Genetic Algorithm," "Application of Genetic Algorithm," "Operators of Genetic Algorithm," "Representation of Genetic Algorithm," and "Variants of Genetic Algorithm" were used throughout searches.

The genetic algorithm research for multimedia applications was also mentioned. All duplicate publications and papers published before 2007 were disregarded while screening research papers. On the basis of submissions from 2007 and duplicates, 4340 research articles were chosen. Following that, 4050 research articles were discarded based on their titles. After reviewing the abstracts of 220 research publications, they were excluded. After the third round of screening, 70 research articles remained. Following thorough examination of the papers and the discovery of facts, 40 further research articles were rejected. The final 30 research articles are chosen for evaluation after the fourth round of screening. The genetic algorithm for multimedia applications, improvements to its genetic operators, and hybridization of the genetic algorithm with other well-known metaheuristic algorithms are all topics covered in the chosen research articles. The previous section outlines the advantages and disadvantages of genetic operators.

Classical Genetic Algorithm

An optimization algorithm that draws its inspiration from natural selection is called a genetic algorithm. It is a population-based search algorithm that makes use of the idea of the strongest surviving. By applying genetic operators repeatedly to people already existing in the population, new populations are created. The main components of a genetic algorithm are chromosomal representation, selection, crossover, mutation, and fitness function computing.

The search procedure is changed constantly by the genetic algorithm using the likelihoods of mutation and crossover. The encoded genes may be changed by genetic algorithms. many people may be evaluated by a genetic algorithm, which can then provide many ideal results. Genetic algorithms are thus superior at doing global searches.

The original schema must be replaced by a changed schema, according to the schema theorem. The new schema preserves the starting population throughout the early stages of evolution in order to retain population variety. The proper genetic scheme will emerge at the conclusion of evolution to avoid any modification of the ideal genetic scheme.

The gene on the chromosome is represented by a tree of functions or instructions in tree encoding. Any programming language may be coupled to these operations and directives. This and the portrayal of repression in tree style are extremely similar. The majority of the time, evolving programs or expressions employ this kind of encoding.

Selection techniques

In genetic algorithms, selection is a crucial phase that decides whether a certain string will take part in reproduction or not. The reproduction operator is another name for the selection stage. The selection pressure affects how quickly GA converges. The popular selection methods include the Boltzmann, rank, tournament, and stochastic universal sampling.

All potential strings are mapped onto a wheel in a roulette wheel selection, and a section of the wheel is allotted to each string based on its fitness value. Then, this wheel is turned at random to choose certain solutions that will take part in the development of the next generation. However, it has a lot of issues, such inaccuracies brought on by its stochastic character. By integrating the idea of determinism in the selection process, De Jong and Brindle changed the roulette wheel selection technique to eliminate mistakes. The modified version of roulette wheel selection is called rank selection. Instead of using fitness value, it uses rankings. They are assigned rankings based on their fitness worth so that each person has a chance to be chosen in accordance with their ranks. Premature convergence of the solution to a local minimum is less likely when using the rank selection approach.

Multi-objective Genetic Algorithms

The updated form of the basic genetic algorithm is the multi-objective genetic algorithm. In terms of fitness function assignment, multi-objective genetic algorithms are distinct from genetic algorithms. Genetic algorithms are reminiscent of the subsequent processes. The primary goal of a multi-objective genetic algorithm is to produce the best Pareto Front possible in the objective space while ensuring that no fitness function can be improved further without affecting the others. The three primary objectives of multi-objective genetic algorithms are convergence, variety, and coverage. The Pareto-based and decomposition-based Multi-objective Genetic Algorithms are the two basic categories into which the Multi-objective Genetic Algorithms are divided. The subsections before this one cover these methods.

Pareto-based multi-objective Genetic Algorithms

Multi-objective Genetic Algorithms introduced the idea of Pareto dominance. The first multi-objective genetic algorithm was created by Fonseca and Fleming. To address multimodal issues, the specialty and decision maker ideas were introduced. Multi-objective genetic algorithms, however, struggle with parameter tuning issues and varying levels of selection pressure. A niched Pareto genetic method was put out by Horn et al. that made use of Pareto dominance and tournament selection. A genetic algorithm for non-dominated sorting was developed by Deb and Srinivas. Its drawbacks include a lack of elitism, the requirement for parameter sharing, and a high level of computational complexity. Deb and colleagues created a quick elitist non-dominated sorting genetic algorithm to solve these issues. NSGA-II's performance might suffer from a variety of real-world issues. The variety in the Pareto-front could not be sustained by NSGA-II. Luo et al. added a dynamic crowding distance to NSGA-II to help with this issue. Creating a Multi-objective Micro Genetic Algorithm was done by Coello and Pulido. The non-dominated solutions were kept in an archive. Many objective challenges may cause Pareto-based techniques to perform worse.

Decomposition-based Multi-objective Genetic Algorithm

Multi-objective genetic algorithms based on decomposition divide the given issue into many smaller problems. The answers to these subproblems are exchanged with those to nearby subproblems as they are addressed concurrently. A multi-objective genetic local search was created by Ishibuchi and Murata. In multi-objective genetic local search, the parents were chosen using random weights, and the offspring was located using local search. They employed a roulette wheel selection technique and generation replacement. The Multi-objective Genetic Local Search was modified by Jaszkiewicz by using several parental selection processes. As an extension of multi-objective genetic algorithms, Murata and Gen suggested a cellular genetic algorithm for multi-objective optimization. They improved Multi-objective Genetic Algorithms by include cellular structure. The selection operator in C- Multi-objective Genetic Algorithms was applied to each cell's neighbors. By adding an immigration process, C- Multi-objective Genetic Algorithms was expanded and renamed CI- Multi-objective Genetic Algorithms. Multi-objective Genetic Algorithms on a decomposition basis were suggested by Patel et al. For the purpose of generating weight vectors, they combined opposition-based learning with D-Multi-objective Genetic Algorithms. D-MOGA can balance the exploration of the search space with a variety of potential solutions.

Parallel Genetic Algorithms

The goal of parallel genetic algorithms is to increase computing efficiency and solution quality by using dispersed people. Master-slave parallel genetic algorithms, fine-grained parallel genetic algorithms, and multi-population coarse-grained parallel GAs are the three basic categories into which parallel GAs fall. The calculation of fitness functions in master-slave parallel genetic algorithms is split across a number of processors. Parallel computers are employed in fine-grained genetic algorithms to tackle practical issues.

The genetic operators are restricted to their immediate surroundings. However, people are free to engage with one another. The exchange of people between sub-populations is carried out via coarse-grained genetic algorithms. During migration, the control settings are also transmitted. Maximizing memory bandwidth and setting up threads to take use of GPU power are the key issues in parallel genetic algorithms.

Master slave parallel Genetic Algorithm

In comparison to other methods, master-slave parallel genetic algorithms make use of a huge number of processors. The number of processors may be raised to increase the calculation of fitness functions. MS-PGA was utilized by Hong et al. to resolve data mining issues. Along with parallel genetic algorithms, fuzzy rules are applied. On slave machines, the fitness function was evaluated. Its drawback is that calculation takes a long time. Sahingzo used MS-PGA to the issue of UAV route finding. On processors, the genetic operators were put to use. Four-core multicore CPUs were used. Slave devices were used for selection and fitness assessments. On the issue of traffic assignment, MS-PGA was used. At National University of Singapore, they solved this issue using thirty processors. A web-based parallel genetic algorithm was created by Yang et al. In a distributed context, they built the master slave version of NSGA-II. The mechanism is complicated, however.

Fine grained parallel Genetic Algorithm

Researchers have been examining fine-grained parallel genetic algorithms' migration rules for the last several decades. For migration frequency, Porta et al. used clock-time, which is independent of generations. They made advantage of static configuration and non-uniform structure. The best option was chosen for migration, and the poorest option was swapped out for it. Adaptive migration frequency was utilized by Kurdi. After 10 succeeding generations, the migration process continues until the results remain unchanged. It employed a dynamic, non-uniform structure. The synchronization of local best solutions to create a global best solution. The most effective global solutions were distributed to all processors for first execution. The number of generations determines the frequency of migration. They made use of a set arrangement and homogenous construction. Parallel GA was utilized by Zhang et al. to resolve the wireless network set cover issue. They divided the population into smaller populations using the divide-and-conquer approach. Following that, local solutions were subjected to genetic operations, and the Kuhn-Munkres algorithm was employed to combine the local solutions.

Coarse grained parallel Genetic Algorithm

A Graph Cell was suggested by Pinel et al. One solution was initiated using the Min-Min heuristic approach, while the population was started using random numbers. The suggested method was implemented on 448 CPUs. However, because to their complexity, coarse-grained parallel genetic algorithms are less often utilized. Multiple applications often employ hybrid parallel GAs. A pool-based Birmingham cluster Genetic Algorithm was suggested by Shayeghi et al. The master node was in charge of overseeing the whole population. Slave node chose the answers from the whole population and carried them out. The calculation is done using 240 processors. To maximize the switching angle of inverters, Roberge et al. adopted a hybrid technique. For the calculation of the fitness function, they utilized four alternative methodologies. The most common hardware nowadays for parallel Genetic Algorithms is the GPU, cloud, and grid.

Chaotic Genetic Algorithms

Premature convergence is Genetic Algorithms' fundamental flaw. To solve this issue, chaotic systems are included into genetic algorithms. Premature convergence is eliminated by the chaotic genetic algorithm's variety. They made use of six distinct chaotic maps. The chaotic Genetic Algorithms developed by Henon, Ikeda, and Logistic performed better than the traditional Genetic Algorithm. These methods do, however, have a substantial computational complexity problem. To solve the local optima issue, Ebrahimzadeh and Jampour employed Lorenz chaotic genetic operators in genetic algorithms. The suggested method, meanwhile, was unable to identify a connection between entropy and chaotic map.

Instead of using a random selection of the beginning population, Javidi and Hosseinpourfard used two chaotic maps, the logistic map and the tent map, to generate chaotic values. The suggested chaotic Genetic Algorithms outperform the GA in terms of performance. However, the great computing cost of this approach is a drawback. Entropy was included by Fuertes et al. into chaotic genetic algorithms. Through chaotic maps, the control settings are changed. They looked at the connection between performance improvement and entropy.

Additionally, multi-objective and hybrid genetic algorithms have employed chaotic systems. For the purpose of tackling bi-level programming issues, Abo-Elnaga and Nasr incorporated chaotic systems into modified genetic algorithms. The suggested algorithm benefits from chaos by reducing local optima and improving convergence.

Hybrid Genetic Algorithms

Image denoising techniques, chemical process optimization, and many other optimization techniques may readily be used with genetic algorithms to improve efficiency. The key benefits of combining Genetic Algorithms with other techniques include improved control parameter settings, greater solution quality, and increased efficiency.

Literature has shown that population size has a significant impact on the capacity of genetic algorithms to sample data. Local search techniques including the memetic algorithm, Baldwinian, Lamarckian, and local search have been combined with genetic algorithms to overcome this issue. With this combination, intensification and diversity are properly balanced. Setting parameters is a further issue with genetic algorithms. Finding the right control settings requires a lot of work. To tackle this issue, Genetic Algorithms may be used with the other metaheuristic methods. The problems stated in the previous subsections have been resolved using hybrid genetic algorithms.

Replacement of genetic operators

The genetic operators stated in Section 3.2 may be swapped out for other search methods. Leng created a directed genetic algorithm that makes use of guided local search penalties. These fines were used to the fitness function of genetic algorithms to enhance their efficiency. Instead of using normal crossover, Headar and Fukushima employed simplex crossover. Simulated annealing took the role of the usual mutation operator. The performance of Genetic Algorithms is enhanced by using the fundamental principles of quantum computing. For the purpose of solving the three-matching issue, genetic algorithms may include the heuristic crossover and hill-climbing operators.

Optimize control parameters

Under order to keep intensification and diversification under check, Genetic Algorithm control parameters are essential. It is possible to estimate the proper control parameters for genetic algorithms using fuzzy logic. Besides this, the control parameters of other approaches may be improved using genetic algorithms. Neutral networks' learning rate, weights, and topology have all been improved using genetic algorithms. To calculate the ideal value of fuzzy membership in a controller, one may utilize genetic algorithms.

Applications

With excellent accuracy rates, genetic algorithms have been used to solve a variety of NP-hard problems. A few application fields have had success with the use of genetic algorithms.

Operation management

GA is an efficient metaheuristic for solving operation management problems such as facility layout problem, supply network design, scheduling, forecasting, and inventory control.

Scheduling

Genetic Algorithms shows the superior performance for solving the scheduling problems such as job-shop scheduling, integrated process planning and scheduling, etc. To improve the performance in the above-mentioned areas of scheduling, researchers developed various genetic representation genetic operators, and hybridized Genetic Algorithms with other methods.

Multimedia

GAs have been applied in various fields of multimedia. Some of well-known multimedia fields are encryption, image processing, video processing, medical imaging, and gaming.

Information security

The development of multimedia apps has allowed for the transmission of photos, videos, and audios via the Internet. The pictures are more mistake prone during transmission, according to research discovered in the literature. As a result, methods of picture protection including encryption, watermarking, and cryptography are needed. The input parameters for encryption are necessary for the traditional image encryption algorithms. A poor choice of input settings will result in insufficient encryption. The right control settings have been chosen using genetic algorithms and their variations. A multi-objective genetic algorithm was created by Kaur and Kumar to optimize the chaotic map's control settings. The beta chaotic map was used to produce the secret key. The picture was encrypted using the generated key. In addition to parallel genetic algorithms, the picture was encrypted.

Image processing

Preprocessing, segmentation, object identification, denoising, and recognition are the primary image processing activities. Image segmentation is a crucial phase in the process of solving image processing issues. Due of their superior search capabilities, Genetic Algorithms are utilized to tackle this issue. Enhancement is a method for raising an image's quality and contrast. The provided picture must have higher image quality for analysis. Images have been enlarged and the natural contrast has been improved using genetic algorithms. To combine the properties of noise and color, several researchers are focusing on hybridizing a rough set with an adaptive evolutionary algorithm. The noise in the provided picture was removed using Genetic algorithms. To denoise the noisy picture, fuzzy logic and genetic algorithms might be combined. The following picture may be restored using a method based on genetic algorithms to get rid of haze, fog, and smog. Real-world object identification and recognition problems are difficult to solve. The Gaussian mixture model performs better throughout the process of detection and recognition. Genetic algorithms are used to optimize the control settings.

Video processing

Pattern recognition and computer vision have both made extensive use of video segmentation. With regard to video segmentation, there are several serious problems. These provide precise borders and help identify objects from the surrounding area. These problems can be solved using genetic algorithms. Chao et al. employed Genetic Algorithms to build genetic algorithms effectively for gesture recognition. They used genetic algorithms and discovered that robot vision had a 95% accuracy rate. In addition to fuzzy classifiers, Kaluri and Reddy suggested an adaptive evolutionary algorithm-based technique for recognizing sign gestures. When compared to the current approach, which has an accuracy rate of 79%, they reported an enhanced recognition rate of

85%. Face recognition, in addition to gesture recognition, is crucial for criminal identification, autonomous vehicles, surveillance, and robotics. The occlusion, orientations, emotions, stance, and lighting conditions may all be handled by genetic algorithms.

Wireless networking

Genetic algorithms have been utilized to address a variety of wireless networking problems because they are adaptable, scalable, and simple to implement. Routing, quality of service, load balancing, localisation, bandwidth allotment, and channel assignment are the major problems with wireless networking. For the purpose of addressing the routing problems, Genetic Algorithms have been hybridized with different metaheuristics. Hybrid genetic algorithms are utilized for load balancing in addition to developing efficient pathways between pairs of nodes.

Localization

The process of determining the location of wireless nodes is called as localization. It plays an important role in disaster management and military services. Yun et al. used Genetic Algorithms with fuzzy logic to find out the weights, which are assigned according to the signal strength. Zhang et al. hybridized Genetic Algorithms with simulated annealing to determine the position of wireless nodes. SA is used as local search to eliminate the premature convergence.

Bandwidth and channel allocation

It's difficult to allocate bandwidth appropriately. The bandwidth allocation issue has been solved using genetic algorithms and its variations. In order to examine bandwidth allocation under QoS restrictions, genetic algorithms were deployed. Resource use, bandwidth distribution, and computing time may all be part of the genetic algorithms' fitness function. The channel allocation in wireless networks is a significant problem. The basic goal of channel allocation is to maximize both the number of channels and frequency reuse at the same time. The channel allocation issue in cognitive radio networks was solved by Friend et al. using distributed island genetic algorithms. For channel assignment, Zhenhua et al. used a modified immune genetic algorithm. They used various immune operators and encoding schemes. For the purpose of solving the static and dynamic channel allocation problems, Pinagapany and Kulkarni created parallel genetic algorithms.

Challenges and future possibilities

In this section, the main challenges faced during the implementation of Genetic Algorithms are discussed followed by the possible research directions.

Challenges

Despite the several advantages, there are some challenges that need to be resolved for future advancements and further evolution of genetic algorithms. Some major challenges are given below:

Selection of initial population

Genetic algorithms are always thought to function best with a certain initial population. The quality of the solution is also impacted by population size. According to the researchers, the algorithm requires more processing time when a huge population is taken into account. The limited population might, however, result in inadequate solutions. Finding the right population size is thus a constant challenge. Harik and Lobo used the self-adaptation approach to study the population. They used two strategies, including

(1) Use of self- adaption prior to execution

of algorithm, in which the size of population remains the same and

(2) In which the self-adaption used during the algorithm execution where the population size is affected by fitness function.

Premature convergence

For Genetic Algorithms, premature convergence is a prevalent problem. It may result in the deletion of alleles, which makes it difficult to pinpoint a gene. Premature convergence is the idea that if the optimization problem coincides too soon, the outcome will be poor. Some studies recommended using variety to overcome this problem. It is important to apply selection pressure to boost variety. The degree of selection pressure favors the superior members of the initial population of genetic algorithms. The population utilizing SP1 should be bigger than the population using SP2 if selection pressure is stronger than some other selection pressure. Higher selection pressure may reduce population variety, which might cause convergence too soon.

Selection of efficient fitness functions

The driving force, or fitness function, is what determines who is the fittest person in each algorithm iteration. A pricey fitness function can be changed if there aren't many iterations. The cost of computing might rise as the number of repetitions grows. The choice of fitness function is based on both its applicability and its computing cost. For the purpose of classifying the texts in, the writers employed the Davies-Bouldin index.

Degree of mutation and crossover

The fundamental components of genetic algorithms are the crossover and mutation operators. There won't be any new information accessible for evolution if the mutation is not taken into account. If crossover is not taken into account during evolution, the algorithm may provide local optimum results. The strength of these operators has a significant impact on how well Genetic Algorithms work. To guarantee the global optima, the right balance between these operators is necessary. The precise degree required for an efficient and ideal solution cannot be determined by the probabilistic nature.

Selection of encoding schemes

A specific encoding strategy is required by genetic algorithms for a certain issue. There is no overarching process for determining if a certain encoding system is appropriate for any kind of real-world issue. Two separate encoding systems are needed if there are two distinct challenges. The encoding systems, according to Ronald, need to be made to be more complex than the redundant forms. The genetic operators should be used in a way that prevents them from favoring duplicate forms.

Conclusions

The organized and explicated perspective of genetic algorithms is presented in this study. Applications of Genetic Algorithms and its variations have been explored. We explore genetic operators that are application-specific. Some genetic operators have a representational purpose. They do not, however, apply to research fields. Numerous studies have been done on the function of genetic operators such crossover, mutation, and selection in preventing early convergence. It has been stated how genetic algorithms and their variations may be used in many study fields. This study focused mostly on wireless network applications and multimedia. The difficulties and problems highlighted in this study will aid practitioners in conducting their studies.

Metaheuristic algorithms and genetic algorithms have various benefits in different study areas.

The purpose of this article is to not only offer the source of current genetic algorithm research, but also to provide information on each genetic algorithm component. It will motivate researchers to comprehend the basics of genetic algorithms and apply their understanding to their research challenges.

References

1. *Abbasi M, Rafiee M, Khosravi MR, Jolfaei A, Menon VG, Koushyar JM (2020) An efficient parallel genetic algorithm solution for vehicle routing problem in cloud implementation of the intelligent trans- portation systems. Journal of cloud Computing 9(6)*
2. *Abdelghany A, Abdelghany K, Azadian F (2017) Airline flight schedule planning under competition. Comput Oper Res 87:20–39*
3. *Abdulal W, Ramachandram S (2011) Reliability-aware genetic scheduling algorithm in grid environment. International Conference on Communication Systems and Network Technologies, Katra, Jammu, pp 673– 677*
4. *Abdullah J (2010) Multi-objectives ga-based QoS routing protocol for mobile ad hoc network. Int J Grid Distrib Comput 3(4):57–68*
5. *Abo-Elnaga Y, Nasr S (2020) Modified evolutionary algorithm and chaotic search for Bilevel program- ming problems. Symmetry 12:767*
6. *Afrouzy ZA, Nasseri SH, Mahdavi I (2016) A genetic algorithm for supply chain configuration with new product development. Comput Ind Eng 101:440–454*
7. *Aiello G, Scalia G (2012) La, Enea M. A multi objective genetic algorithm for the facility layout problem based upon slicing structure encoding Expert Syst Appl 39(12):10352–10358*
8. *Alaoui A, Adamou-Mitiche ABH, Mitiche L (2020) Effective hybrid genetic algorithm for removing salt and pepper noise. IET Image Process 14(2):289–296*
9. *Alkhafaji BJ, Salih MA, Nabat ZM, Shnain SA (2020) Segmenting video frame images using genetic algorithms. Periodicals of Engineering and Natural Sciences 8(2):1106–1114*
10. *Al-Oqaily AT, Shakah G (2018) Solving non-linear optimization problems using parallel genetic algo- rithm. International Conference on Computer Science and Information Technology (CSIT), Amman, pp. 103–106*
11. *Alvesa MJ, Almeidab M (2007) MOTGA: A Multi-objective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. Comput Oper Res 34:3458–3470*
12. *Arakaki RK, Usberti FL (2018) Hybrid genetic algorithm for the open capacitated arc routing problem. Comput Oper Res 90:221–231*
13. *Arkhipov DI, Wu D, Wu T, Regan AC (2020) A parallel genetic algorithm framework for transportation planning and logistics management. IEEE Access 8:106506–106515*
14. *Azadeh A, Elahi S, Farahani MH, Nasirian B (2017) A genetic algorithm-Taguchi based approach to inventory routing problem of a single perishable product with transshipment. Comput Ind Eng 104:124– 133*
15. *Baker JE, Grefenstette J (2014) Proceedings of the first international conference on genetic algorithms and their applications. Taylor and Francis, Hoboken, pp 101–105*
16. *Bolboca SD, JAntschi L, Balan MC, Diudea MV, Sestras RE (2010) State of art in genetic algorithms for agricultural systems. Not Bot Hort Agrobot Cluj 38(3):51–63*
17. *Bonabeau E, Dorigo M, Theraulaz G (1999) Swarm intelligence: from natural to artificial systems. Oxford University Press, Inc*
18. *Burchardt H, Salomon R (2006) Implementation of path planning using genetic algorithms on Mobile robots. IEEE International Conference on Evolutionary Computation, Vancouver, BC, pp 1831–1836*
19. *Burdsall B, Giraud-Carrier C (1997) Evolving fuzzy prototypes for efficient data clustering," in second international ICSC symposium on fuzzy logic and applications. Zurich, Switzerland, pp. 217-223.*
20. *Burkowski FJ (1999) Shuffle crossover and mutual information. Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 1999, pp. 1574–1580*
21. *Chaiyaratana N, Zalzala AM (2000) "Hybridisation of neural networks and a genetic algorithm for friction compensation," in the 2000 congress on evolutionary computation, vol 1. San Diego, USA, pp 22–29*
22. *Chen R, Liang C-Y, Hong W-C, Gu D-X (2015) Forecasting holiday daily tourist flow based*

on seasonal support vector regression with adaptive genetic algorithm. Appl Soft Comput 26:434–443

23. J.R. Cheng and M. Gen (2020) Parallel genetic algorithms with GPU computing. Impact on Intelligent Logistics and Manufacturing.

24. Cheng H, Yang S (2010) Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. Applications of evolutionary computation. Springer, In, pp 562–571

25. Cheng H, Yang S, Cao J (2013) Dynamic genetic algorithms for the dynamic load balanced clustering problem in mobile ad hoc net-works. Expert Syst Appl 40(4):1381–1392

26. Chouhan SS, Kaul A, Singh UP (2018) Soft computing approaches for image segmentation: a survey. Multimed Tools Appl 77(21):28483–28537

27. Chuang YC, Chen CT, Hwang C (2016) A simple and efficient real-coded genetic algorithm for constrained optimization. Appl Soft Comput 38:87–105

28. Coello CAC, Pulido GT (2001) A micro-genetic algorithm for Multi-objective optimization. In: EMO, volume 1993 of lecture notes in computer science, pp 126–140. Springer

29. Das, K. N. (2014). Hybrid genetic algorithm: an optimization tool. In global trends in intelligent computing Research and Development (pp. 268-305). IGI global.

30. Das AK, Pratihar DK (2018) A direction-based exponential mutation operator for real-coded genetic algorithm. IEEE International Conference on Emerging Applications of Information Technology.

31. Dash SR, Dehuri S, Rayaguru S (2013) Discovering interesting rules from biological data using parallel genetic algorithm, 3rd IEEE International Advance Computing Conference (IACC), Ghaziabad,, pp. 631– 636.

32. Datta D, Amaral ARS, Figueira JR (2011) Single row facility layout problem using a permutation-based genetic algorithm. European J Oper Res 213(2):388–394

33. de Ocampo ALP, Dadios EP (2017) "Energy cost optimization in irrigation system of smart farm by using genetic algorithm," 2017IEEE 9th international conference on humanoid. Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Manila, pp 1–7

34. Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. Complex Systems 9: 115–148

35. Deb K, Deb D (2014) Analysing mutation schemes for real-parameter genetic algorithms. International Journal of Artificial Intelligence and Soft Computing 4(1):1–28

36. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist Multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

37. Deep K, Das KN (2008) Quadratic approximation based hybrid genetic algorithm for function optimiza- tion. Appl Math Comput 203(1):86–98

38. Deep K, Thakur M (2007) A new mutation operator for real coded genetic algorithms. Appl Math Comput 193:211–230

39. Deep K, Thakur M (2007) A new crossover operator for real coded genetic algorithms. Appl Math Comput 188:895–911

40. Dhal KP, Ray S, Das A, Das S (2018) A survey on nature-inspired optimization algorithms and their application in image enhancement domain. Archives of Computational Methods in Engineering 5:1607– 1638