EAI Endorsed Transactions
on Scalable Information System

Research Article **EAI.EU**

Comparative analysis of various Evolutionary Algorithms: Past three decades

A. Srikumar[1, *], Sagar Dhanraj Pande[1]

[1] School of Computer Science & Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

Abstract

INTRODUCTION: The Evolutionary algorithms created back in 1953, have gone through various phases of development over the years. It has been put to use to solve various problems in different domains including complex problems such as the infamous problem of Travelling Salesperson (TSP).

OBJECTIVES: The main objective of this research is to find out the advancements in Evolutionary algorithms and to check whether it is still relevant in 2023.

METHODS: To give an overview of the related concepts, subdomains, pros, and cons, the historical and recent developments are discussed and critiqued to provide insights into the results and a better conception of the trends in the domain.

RESULTS: For a better perception of the development of evolutionary algorithms over the years, decade-wise trend analysis has been done for the past three decades.

CONCLUSION: Scope of research in the domain is ever expanding and to name a few EAs for Data mining, Hybrid EAs are still under development.

Keywords: optimization, Evolutionary Algorithms, Genetic Algorithms, Trend Analysis of Genetic algorithms

Received on 04 September 2023, accepted on 02 November 2023, published on 10 November 2023

doi: 10.4108/eetsis.4356

[1]Corresponding author. Email: srikumar293official@gmail.com

1. Introduction

In our quest to understand the world and manipulate it for survival, we encounter various problems some of which require us to find the optimal solution, and others stress learning. Some of the problems of the kind Optimization can be solved using multivariable calculus. The problems of kind combinatorial optimization cannot be solved using calculus as these problems' parameters are not differentiable. Similarly, in the case of multi-objective optimization problems, there is no such point/vector in the defined domain. Evolutionary algorithms are learning-based algorithms that learn from the genotype of the solution and use it to construct a better solution. This character of the Evolutionary algorithm allows us to use it to optimize combinatorial, multi-objective, and other optimization problems.[2]

As EAs are primarily used to solve learning-based problems, it is effectively termed machine learning algorithm as in Figure 1. Over the years EAs have become a popular tool for search, learning, and optimization problems [3]. The Evolutionary algorithms have different evolutionary strategies out of which the crossover and mutation strategy also known as the Genetic algorithm finds itself useful in a wide range of domains [4]. In this paper, we focus on different types of Evolutionary algorithms and their applications, decade-wise trends in the domain of Evolutionary algorithms, their analysis, and future scope for research. The criteria for evaluating the trend are based on the number of legible publications on the topic available on Google Scholar, IEEE explores, Science Direct, Research Gate, and Scopus.

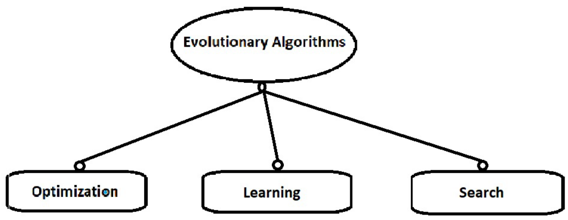

Figure 1: Classification of applications of EA

2. Evolutionary Algorithms

Artificial intelligence is a domain of computer science in which we humans try to create a program/algorithm capable of displaying near-human intelligence. In the process of making an artificially intelligent machine, the researchers started with preliminary human traits and strategies and tried implementing them in a computer program. Though these strategies were efficient in solving one problem, their efficiency and relevance decreased when tested on other problems. In search of answers, we came back to ourselves to understand the thought process behind approaching and solving a problem. It was found that we not only rely on a single strategy but on various others which involve random guessing, blind search, propositional logic, etcetera. Apart from various strategies we develop new strategies and perturb the existing strategies by learning from the past for optimistic results. As the algorithm/strategy for solving problems is not static and changes concerning the experience gained by the agent, these algorithms are called Evolutionary algorithms. This view of problem-solving is different from the knowledge-based/informed solution finder as the latter's knowledge base is dynamic rather in the case of the former, both the knowledge and algorithm are dynamic.

The changes made in the algorithms are dependent on a metaheuristic function known as the Fitness function which is a measure of the performance of the particular version of the algorithm. When a problem is approached, various existing algorithms' fitness is evaluated and then biological mechanisms such as reproduction, mutation, crossover, selection, etcetera are simulated on the selected candidate. The candidate is usually the fittest in the population, the selection criteria are based on the infamous Darwinian theory of evolution "Survival of the fittest". The new candidate in the population is expected to be better than the parent candidate else the best candidate from the population will be chosen as evolution is not always progressive as sometimes recessive genes/characters too surfaces in some candidates. These processes provide the algorithm with a multitude of population with varying characters which is desirable in optimization problems where the goal is to find the global optimum masked by a large number of local optima. [1]

The subdomains of evolutionary algorithms include:

- Genetic algorithm.
- Genetic Programming.
- Evolution Strategy.
- Differential Evolution.
- Neuro Evolution.
- Evolutionary Programming.

Though there are 6 types of Evolutionary algorithms, but it is difficult to differentiate them in the present. During the initial stages they were disjoint but with advancement in research and more exploration the fine line separating these sub types has vanished.[12]

2.1. Genetic algorithms

Genetic Algorithms or the GAs are population-based search algorithms based on the principles of natural selection proposed by J.H. Holland in 1992. The GA's implements the Darwinian principle in selection. The basic structure of a genetic algorithm is discussed below:

1. The vital information is encoded into a gene string.
2. The genetic operations like crossover and mutation are carried out on this gene string by operators called as genetic operators.
3. The genetic operators are iteratively used upon the genotype (String) until an optimal solution candidate is found.

The resultant optimal solution is very near to the global optimum as it is the best candidate of the whole population. This is the reason for GA's success in optimization and learning problems. There is a possibility in which undesirable genotype is transmitted to the descendants, but it won't affect the solution as it perishes in the iterative process (can be proved by Darwinian theory of evolution).[5]

2.1.1 Genetic Strings [5,7]

The Genetic data which is termed as chromosome is represented in GA's using string of primitive data types. The template of describing a subset of chromosome is known as schema. There are various schemes available for chromosome encoding out which Binary encoding is the most popular one. The various schemes are:

1. Binary Encoding: The Genetic string consists of either 0's or 1's as atomic elements. This is the most common type of encoding and has been widely used in GA's. Example of a binary coded string is shown in Figure 2.

1	0	1	1	0	1	1	1	1	1

Figure 2: Binary encoded string

2. Octal Encoding: This type of encoding makes use of numbers belonging to the group modulo 8(Z8=0,1,....,7) as atomic elements. Example of an octal coded string is shown in Figure 3.

1	0	5	7	3	2	1	4	4	1

Figure 3: octal encoded string

3. Hexadecimal Encoding: This type of encoding using uses numbers belonging to the group of modulo 16 (Z16= 0, 1, 2, 3, ...,F). In contrast to using numbers it also includes alphabets to represent numbers greater than 9. Example of a Hexadecimal coded string is shown in Figure 4.

1	A	5	B	F	C	1	4	4	F

Figure 4: hexadecimal encoded string

4. Permutation Encoding: In the Genetic string encoded using permutation encoding, every atomic element represents a sequence or position. Hence finds itself applicable for ordering based problems like the TSP (Travelling sales problem).

5. Value Encoding: This type of encoding uses Strings to represent values in a chromosomal string. The value represented by the string can be integers, floats, character, etc. Value encoding is useful in modelling neural networks Example of a value coded string is shown in Figure 5.

B	A	G	B	F	C	C	T	T	F

Figure 5: value encoded string

6. Tree Encoding: In lieu of using an array to visualize a genetic string, Tree encoding uses a tree data structure to do the same.

2.1.2 Genetic Strings

The two important genetic operations are Crossover and mutation.

Crossover is a genetic operation which is inspired from the process of recombination in reproduction. Essentially the gene of parents is combined in a certain fashion to get an entirely new genotype/individual [8]. There are several types of crossover operators available, among them the most famous types are mentioned and explained below:

1. Single Point Crossover: This type of crossover operator randomly chooses a point in the gene string of one parent and combines it with another half obtained from the other parent. [9,5]

2. N- point Crossover: It is similar to the single point of crossover except for the point that in lieu of choosing a random single point it chooses N points.

3. Uniform Crossover: This variety of crossover technique considers each gene as a separate quantity and the genes are randomly chosen from the two parents.[5]

4. Partially matched Crossover (PMX): Of the two chosen parents, one parent donates some part of the gene string to the other parent and then the left-out alleles are matched with its counterpart in the string and replaced such that no repetition/duplication is left.

5. Order crossover: This method of implementing crossover selects sub string from the parents and unlike the PMX it fills the left-over alleles according to a predefined order. [8,5]

6. Shuffle crossover: This method of crossover induces randomness in crossover as the other crossover techniques to have a fixed pattern hence reducing the availability of options in the new generation. It shuffles the genetic string before combination and unshuffles them after combination. It has a new efficient variant namely Reduced surrogate crossover (RCX).[5]

7. cycle crossover: This method of crossover chooses genes from one parent and then the other in cycles in such a way that no collision occurs, and the position of the gene is preserved. [9,5]

Mutation is a genetic operation which changes certain genes of the child/young ones randomly. This phenomenon occurs usually due to error while copying of genes in reproduction, this phenomenon is simulated using random function. This operation ensures diversity in a population. The changes induced by mutation can be beneficial or recessive. Some of the Mutation operators are:

1. Displacement mutation: A sub string of the main gene string is displaced from its original position such that the obtained combination is legal.

2. Simple inversion: This method of mutation inverts the string of gene present given between two randomly chosen points.[5]
3.

3 Power mutation: This method of mutation was proposed in 2007 in [67]. The distribution function:

4.

$$f(x) = a * x^b$$

where b=a-1 and a is a random power that the user chooses.

5.

The density function:

$$F(x) = x^p$$

The mutated string z:

$$z = mean(x) - d(mean(x) - xl)$$

for Y<r

$$z = mean(x) - d(x^u - mean(x))$$

for Y>=r

$$Y = mean(x) - x^l/x^u - x^l$$

where x^l is the upper limit of the string operated on and x^u
is the lower limit of the string operated on.[67]

2.1.3 Choice of Genetic Operation

The strategy of using crossover and mutation operator is an important aspect of GA's. This strategy determines the diversity in the next generation population. There two different theories with respect to devising this strategy. One supports mutation and the other supports crossover and views mutation as a secondary operator.

Fogel and Atmar theory: According to the research done by L. Fogel (1966) which was later continued by Schaffer and concluded by D.Fogel and Atmar(1996), Crossover doesn't have advantage over mutation. Which basically means both crossover and mutation are equally important.They can be used dynamically.

Holland's theory: Holland's Theory (1975) focuses more on crossover
being a powerful operator and mutation being a sideline operator.
Though both theories have been researched by scholars and proved by different methods there is no theoretical justification for either of the two contradicting theories. [10]

2.2. Genetic Programming

Genetic Programming is a population-based problem solver. It is similar to GAs in terms of functionality and algorithm. It too uses genetic operators as crossover and mutation. Unlike GA's Genetic programming has variable gene string length. This adds to the flexibility of the algorithm. This feature of Genetic programming is realized by the use syntax tree data structure to represent code/solution candidate.

2.2.1 Gene String Representation by Syntax Tree

The tree includes nodes(points) and relations(links). Every internal node is called as a function, and the leaf nodes are called as terminals. The nodes consist of instructions and the links consist of argument. The architecture of genetic program is defined by the attributes of this syntax tree.

2.2.2 Linear Genetic Programming

Linear GPs are special sub type of GP's that makes use of functional expressions in place of trees as the authentic GP does. This expression is evolved further to get offspring for the next generation of the population. The LGPs were previously implemented using Lisp (a functional programming language) but are now implemented using an imperative programming language like C. [62,63]

2.3. Evolution Strategy

Evolution Strategy algorithms are population-based algorithms which are solely based on recombination (crossover/reproduction) and mutation. Both the process of mutation and crossover has been discussed earlier under genetic algorithms. The only difference between GA's and ES (evolution strategy) is, GA makes use of ES and other genetic operators. We can essentially call GA as super set of all other sub variants of evolutionary algorithms.[12]

2.4. Differential Evolution

Differential Evolutionary algorithms are stochastic population-based algorithms that is implemented on a continuous data set. It was created to optimize variables be it continuous or discrete [65]. It is faster and simpler than many of its counter parts. The gene string of DE's represents a vector. Every member of the population is represented/ identified by this vector. The fitness function, crossover, mutation and other genetic operations are implemented using differential calculus operations like partial differentiation and multi variable differentiation.

2.5. Neuro Evolution

Neuro evolution is an evolutionary algorithm which implements the concept of neural networks to solve variety of problems which includes optimization and image detection. The neural network consists of individual units known as neuron. Each neuron has fibrous root like structure called dendrites which receives input information/data in the form of electrical impulses. In an actual neuron this data input is processed in the cell body and then the output is given through the axon terminal of the neuron. This is the mainspring essence of data transmission by a neuron.

A neuron in Neuro evolution is realised by directed graphs where vertices represent the data states and edges represent the relation/synaptic weights of the particular data input. The effect of each input on the output is controlled by their respective synaptic weights. [1,14]

The Artificial neuron also known as Perceptron is diagrammatically represented in Figure 6.

2.5.1 Evolving a Neural Network

Neural network as it is, is not suitable for solving real world problems as we need to configure the synaptic weights for every synaptic node and the number of independent variables/inputs also changes from one scenario to the next. To solve real world problems, we require a robust neural network. This requirement is fulfilled by the neuro evolution algorithm as it evolves the neural network for a variety of parameters with varied inputs, which results in production of a population of a variety of neural networks varying in synaptic weights. Based on the performance of individual neural networks on the validation data the best neural network is chosen, and its synaptic weights are recorded.[14]

2.5. Neuro Evolution

Evolutionary programming is generated and test algorithm which solely depends on mutation and selection to generate optimal solution. An offspring is generated using different mutation operators and based upon their fitness the fittest offspring is chosen and this process is repeated until global optimum is reached.[13]

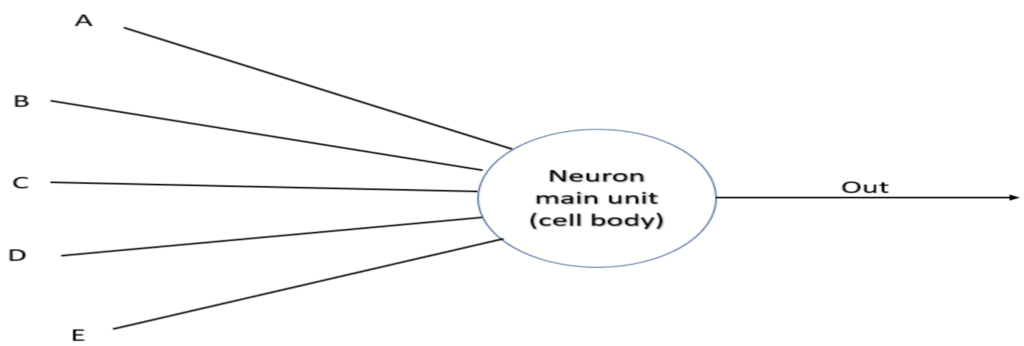

Figure 6. Neuron Representation
w1 ∗ A + W 2 ∗ B + W 3 ∗ C + W 4 ∗ D + W 5 ∗ E = Output
where w1, w2, w3, w4 and w5 are weights assigned to each input node

3. Decade wise Trend

3.1. 1990-2000

In the Early 1990's the idea of Evolutionary algorithms had already been established and was a common algorithm for solving Optimization problems. At that period of time optimization was the popular problem which the researchers tried to solve using different evolutionary algorithms. According to researchers Evolutionary Algorithms as a field of research attained its maturity during the mid-1990s [21]. This statement can be proven right owing to evidence that researchers tried to prune the EAs to get better results and tried using them for various real world engineering problems like the truss tree, unit commitment and Stacking sequence design [26]. In the late 1990s the research then was oriented to development of hybrid Evolutionary algorithms which offered better accuracy and could serve as a general evolutionary algorithm [24,29,31].

* Figure 7 depicts the distribution of highly cited research papers published during the time period of 1990-2000.
* Tab: Timeline 1990-2000 has the sorted works of this timeline. 3.2. 2001-2010

The early part of decade saw surge in the number of researches regarding variety of applications that EA's can be put into. The generalized Evolutionary algorithms were developed during this period of time. Topic like Distributed EA's surfaced in this decade [34]. Hybridisation of Evolutionary algorithms and application of the same on Data mining became quite famous research field in the later part the decade. It is visible from the collected that application and future scope/proposals had increased as compared to the previous decade.

* Figure 8 depicts the distribution of highly cited research papers published during the time period of 2001-2010. Tab: Timeline 2001-2010 has the sorted works of this timeline.

3.3. Neuro Evolution

This decade saw a rise in number of survey papers in the domain of Evolutionary Algorithms which can be accounted to the decades of research in the domain. The extensive research in evolutionary algorithms not only opened up new domains like Quantum inspired EA but also played an important role in maturing of EA. Even after attaining maturity EA has lot of scope for research as with every research a new possibility/dimension is unlocked. Neuro evolution is the sub field in EA which has attained its prime in this decade [51]. There is significant decline in the evaluation of EA part. This observation can be reasoned by stating that new fields in EA are being researched which can be considered a cooling off period for development in evaluation of EA's. With introduction of new EA’s, the existing definition of fitness may change which in turn will arise the requirement of new fitness functions and hence its assessment also will differ.

* Figure 9 depicts the distribution of highly cited research papers published during the time period of 2011-2022.

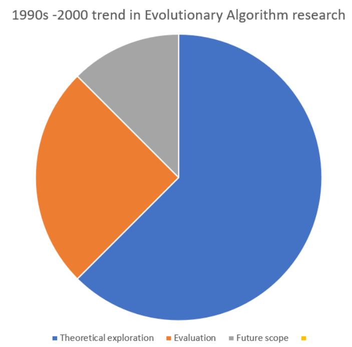

Figure 7: The Trend of timeline 1990s-2000

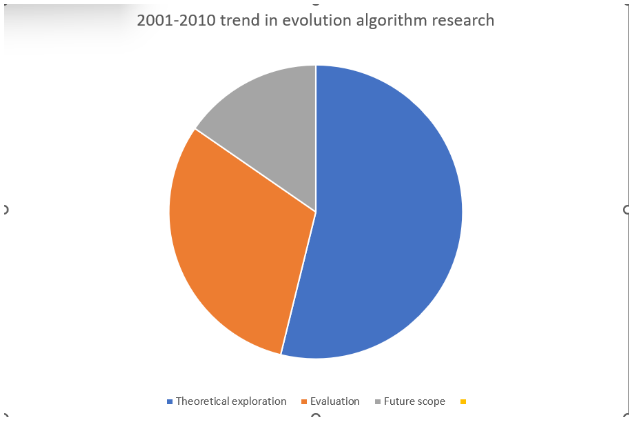

Figure 8: The Trend of timeline 2001s-2010

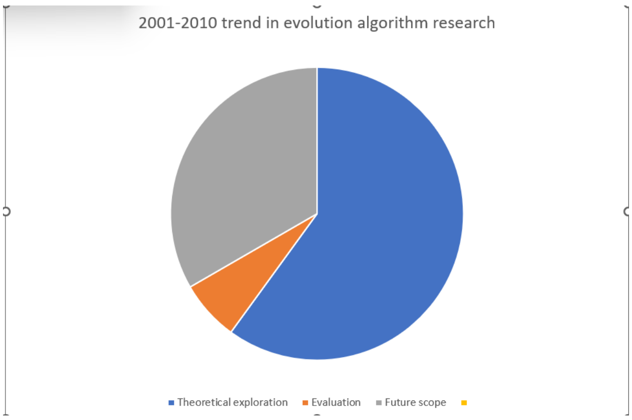

Figure 9: The Trend of timeline 2011s-2023

S.NO	Ref.Id	Name	Purpose
		Theoretical exploration	
1	[17]	An overview of evolutionary algorithms for parameter optimization[17]	GA, ES, EP
2	[19]	Parameter control in evolutionary algorithms[19]	Optimization
3	[21]	Genetic and evolutionary algorithms come of age[21]	Development
4	[22]	Evolutionary algorithms—an overview[22]	General EA
5	[23]	An overview of evolutionary algorithms in multiobjective optimization[23]	Optimization
6	[26]	Evolutionary algorithms for constrained engineering problems[26]	application Stacking sequence design, truss tree,unit commitment
7	[27]	Evolutionary algorithms and gradient search: similarities and differences[27]	comparative analysis
8	[28]	Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art[28]	MO EA analytical insights
9	[30]	Global optimisation by evolutionary algorithms[30]	GA, ES and EP
10	[31]	Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach[31]	SPEA
		Evaluation	
1	[18]	Evaluating evolutionary algorithms[18]	Hill climber
2	[20]	A Comparison of Selection Schemes Used in Evolutionary Algorithms[20]	Selection schema
3	[25]	Multiobjective optimization using evolutionary algorithms[25]	Comparison
4	[26]	Comparison of Multiobjective Evolutionary Algorithms: Empirical Results[26]	comparative analysis of EA based on Optimization
		Future Scope(build up)	
1	[24]	Towards hybrid evolutionary algorithms[24]	Hybridisation of existing EA
2	[29]	EVOLUTIONARY ALGORITHMS IN MACROECONOMIC MODELS[29]	Application based

Table 1: Timeline 1990-2000

S.NO	Ref.Id	Name	Purpose
		Theoretical exploration	
1	[32]	An overview of evolutionary algorithms: practical issues and common pitfalls[32]	GA, ES, Gp and Ep
2	[33]	Parameter control in evolutionary algorithms[33]	parametric control and ES
3	[34]	Introduction to evolutionary algorithms[34]	Distributed EA
4	[35]	Theoretical Aspects of Evolutionary Algorithms[35]	Study of EA w.r.t Randomized search heuristics
5	[40]	Representations for Genetic and Evolutionary Algorithms[40]	GEA-basics
6	[41]	Hybrid Evolutionary Algorithms: Methodologies, Architectures, and Reviews[41]	Hybridisation of EA's
7	[45]	Fast evolutionary algorithms[45]	evolutionary optimization
		Evaluation	
1	[39]	Structural optimization using evolutionary algorithms[39]	GA, ES
2	[43]	Comparing parameter tuning methods for evolutionary algorithms[43]	Optimizing parameters
3	[44]	Performance scaling of multi-objective evolutionary algorithms[44]	comparing MOEA like NSGA-II, SPEA2, PESA
4	[52]	An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems[52]	Classification
		Future Scope(build up)	
1	[36]	parallelism and evolutionary algorithms[36]	Parallel EA's
2	[37]	Diversity-Guided Evolutionary Algorithms[37]	solves population diversity issue using DGEA
3	[38]	Data mining and knowledge discovery with evolutionary algorithms[38]	Application-data mining
4	[42]	A survey of evolutionary algorithms for data mining and knowledge discovery[42]	Survey of EA's for data mining

Table 2: Timeline 2001-2010

S.NO	Ref.Id	Name	Purpose
		Theoretical exploration	
1	[46]	Evolutionary algorithms: A critical review and its future prospects[46]	general EA
2	[47]	Evolutionary Algorithms[47]	Spatial and temporal data mining
3	[48]	Parameter Control in Evolutionary Algorithms: Trends and Challenges[48]	Survey
4	[49]	Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction[49]	Optimization
5	[53]	A survey of evolutionary algorithms for decision-tree induction[53]	Survey
6	[54]	Exploration and exploitation in evolutionary algorithms: A survey[54]	survey
7	[56]	Model-based evolutionary algorithms: a short survey[56]	survey
8	[57]	Analyzing convergence performance of evolutionary algorithms: A statistical approach[57]	statistical analysis of EA
9	[58]	Evolutionary algorithm parameters and methods to tune them[58]	parametric optimization
10	[66]	Evolutionary algorithms and their applications to engineering problems[66]	Applications of EAs
		Evaluation	
1	[60]	A chess rating system for evolutionary algorithms: A new method[60] for the comparison and ranking of evolutionary algorithms	New comparison methodolgy
		Future Scope/Proposal	
1	[50]	Evolutionary algorithms and their applications to engineering problems[50]	application in real life problems
2	[51]	Evolutionary algorithms and neural networks[51]	applications of ANNs
3	[55]	Quantum-inspired evolutionary algorithms: a survey and empirical study[55]	Quantum inspired EA[55]
4	[59]	Evolutionary algorithms in management applications[59]	management applications
5	[61]	Transfer learning based evolutionary algorithm for composite face sketch recognition[61]	facial recognition

Table 3: Timeline 2011-2022

4. Conclusion

From the research, we conclude that the domain of Evolutionary algorithms was well established in the 1990s-2000 period which researchers regarded as the maturity period of Evolutionary algorithms. But it was not the saturation point for the EAs as in the period researchers put EAs in use to solve different problems like data mining. The hybridization of the algorithm was a hot topic among researchers in the mid-2001-2010 period. The 2011-2022 period saw a rise in the use of EAs in Deep learning. This period saw a rise in literature surveys on EAs which conveys that EAs are still being intensively researched and there is still scope to find new applications for EAs. The hybridization of EAs can open up many more fields for research.

5. Future Scope

The Evolutionary algorithm as a research field is in its improvisation and finding wider applications stage. Research in the future will be based on the following:

1. **Developing of a new variety of EA:** Over the years the proposals of new algorithms that are inspired from Genetics have reduced significantly due to exhaustive implementation of these algorithms. Though there is scope for developing a new algorithm inspired from animals similar to the whale optimization algorithm inspired from the method of hunting used by humpback whales [64].

2. **Hybridisation of the existing EA:** The Evolutionary algorithms evolve the population of the solutions that doesn't mean that the algorithm evolves to be a better version of its own. This work had been taken up by the researchers in early 2000s when researchers thought EAs came to a period of maturity. But with development of new algorithms and new discovery in the field of genetics always extends the scope for researchers to hybridize the respective EA or Eas (plural).
3. **Extending applications of EA to new domains:** like games, Cryptocurrencies, Stock market, etc.
4. **Finding more effective evaluation methodology.**

6. References

[1] A First Course in Artificial Intelligence, Deepak Kheemani (2013).
[2] Introduction to Evolutionary Algorithms, Xinjie Yu, Mitsuo Gen.
[3] An overview of evolutionary algorithms: practical issues and common pitfalls, Darrell Whitney
[4] Genetic Programming: A Paradigm For Genetically Breeding Computer Population of Computer Programs to Solve Problems, J.Koza, MIT Press, Cambridge, MA (1992)
[5] Katoch, S., Chauhan, S.S. and Kumar, V. A review on genetic algorithm: past, present, and future. Multimed Tools Appl 80, 8091–8126 (2021). https://doi.org/10.1007/s11042-020-10139-6
[6] Forrest, S., 1996. Genetic algorithms. ACM Computing Surveys (CSUR), 28(1), pp.77-80.
[7] Kumar, A. (2013). Encoding schemes in genetic algorithm. International Journal of Advanced Research in IT and Engineering, 2(3), 1-7.
[8] Lee, S., 2002. Genetic algorithms. Orthogonal arrays.
[9] Kora, P. and Yadlapalli, P., 2017. Crossover operators in genetic algorithms: A review. International Journal of Computer Applications, 162(10).
[10] Spears, W.M., 1993. Crossover or mutation?. In Foundations of genetic algorithms (Vol. 2, pp. 221-237). Elsevier.
[11] Koza, J.R. and Poli, R., 2005. Genetic programming. In Search methodologies (pp. 127-164). Springer, Boston, MA.[12] Beyer, H.G., 2001. The theory of evolution strategies. Springer Science Business Media.
[13] Xin Yao, Yong Liu and Guangming Lin, "Evolutionary programming made faster," in IEEE Transactions on Evolutionary Computation, vol. 3, no. 2, pp. 82-102, July 1999, doi: 10.1109/4235.771163.
[14] Floreano, D., D¨urr, P. and Mattiussi, C., 2008. Neuroevolution: from architectures to learning. Evolutionary intelligence, 1(1), pp.47-62.
[15] A. K. Qin, V. L. Huang and P. N. Suganthan, "Differential Evolution Algorithm With Strategy Adaptation for Global Numerical Optimization," in IEEE Transactions on Evolutionary Computation, vol. 13, no. 2, pp. 398-417, April 2009, doi: 10.1109/TEVC.2008.927706.
[16] Mallipeddi, R., Suganthan, P.N., Pan, Q.K. and Tasgetiren, M.F., 2011. Differential evolution algorithm with ensemble of parameters and mutation strategies. Applied soft computing, 11(2), pp.1679-1696.
[17] B¨ack, T. and Schwefel, H.P., 1993. An overview of evolutionary algorithms for parameter optimization. Evolutionary computation, 1(1), pp.1-23.
[18] Whitley, D., Rana, S., Dzubera, J. and Mathias, K.E., 1996. Evaluating evolutionary algorithms. Artificial intelligence, 85(1-2), pp.245-276.
[19] A. E. Eiben, R. Hinterding and Z. Michalewicz, "Parameter control in evolutionary algorithms," in IEEE Transactions on Evolutionary Computation, vol. 3, no. 2, pp. 124-141, July 1999, doi: 10.1109/4235.771166.
[20] T.Blickle and L. Thiele, "A Comparison of Selection Schemes Used in Evolutionary Algorithms," in Evolutionary

Computation, vol. 4, no. 4, pp. 361-394, Dec. 1996, doi: 10.1162/evco.1996.4.4.361.
[21] Goldberg, David E. ”Genetic and evolutionary algorithms come of age.” Communications of the ACM, vol. 37, no. 3, Mar. 1994, pp. 113+. Gale Academic OneFile, link.gale.com/apps/doc/A15061357/AONE?u=anon 5269f96e sid=googleScholarxid=4c409f9a. Accessed 13 Dec. 2022.
[22] Dasgupta, D. and Michalewicz, Z., 1997. Evolutionary algorithms—an overview. Evolutionary Algorithms in Engineering Applications, pp.3-28.
[23] Fonseca, C.M. and Fleming, P.J., 1995. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary computation, 3(1), pp.1-16.
[24] Preux, P. and Talbi, E.G., 1999. Towards hybrid evolutionary algorithms. International transactions in operational research, 6(6), pp.557-570.
[25] Zitzler, E. and Thiele, L., 1998, September. Multiobjective optimization using evolutionary algorithms—a comparative case study. In International conference on parallel problem solving from nature (pp. 292-301). Springer, Berlin, Heidelberg.
[26] Michalewicz, Z., Dasgupta, D., Le Riche, R.G. and Schoenauer, M., 1996. Evolutionary algorithms for constrained engineering problems. Computers Industrial Engineering, 30(4), pp.851-870.
[27] R. Salomon, ”Evolutionary algorithms and gradient search: similarities and differences,” in IEEE Transactions on Evolutionary Computation, vol. 2, no. 2, pp. 45-55, July 1998, doi: 10.1109/4235.728207.
[28] Van Veldhuizen, D.A. and Lamont, G.B., 2000. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. Evolutionary computation, 8(2), pp.125-147.
[29] Arifovic, J., 2000. Evolutionary algorithms in macroeconomic models. Macroeconomic Dynamics, 4(3), pp.373-414.
[30] Xin Yao, ”Global optimisation by evolutionary algorithms,” Proceedings of IEEE International Symposium on Parallel Algorithms Architecture Synthesis, 1997, pp. 282-291, doi: 10.1109/AISPAS.1997.581678.
[31] Zitzler, E. and Thiele, L., 1999. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE transactions on Evolutionary Computation, 3(4), pp.257-271.
[32] Whitley, D., 2001. An overview of evolutionary algorithms: practical issues and common pitfalls. Information and software technology, 43(14), pp.817-831.
[33] Eiben, A.E., Michalewicz, Z., Schoenauer, M. and Smith, J.E., 2007. Parameter control in evolutionary algorithms. In Parameter setting in evolutionary algorithms (pp. 19-46). Springer, Berlin, Heidelberg.
[34] Streichert, F., 2002. Introduction to evolutionary algorithms. paper to be presented Apr, 4.
[35] Wegener, I., 2001, July. Theoretical aspects of evolutionary algorithms. In International Colloquium on Automata, Languages, and Programming (pp. 64- 78). Springer, Berlin, Heidelberg.
[36] E. Alba and M. Tomassini, ”Parallelism and evolutionary algorithms,” in IEEE Transactions on Evolutionary Computation, vol. 6, no. 5, pp. 443-462, Oct. 2002, doi: 10.1109/TEVC.2002.800880.
[37] Ursem, R.K., 2002, September. Diversity-guided evolutionary algorithms. In International Conference on Parallel Problem Solving from Nature (pp. 462- 471). Springer, Berlin, Heidelberg.
[38] Freitas, A.A., 2002. Data mining and knowledge discovery with evolutionary algorithms. Springer Science Business Media.
[39] Lagaros, N.D., Papadrakakis, M. and Kokossalakis, G., 2002. Structural optimization using evolutionary algorithms. Computers structures, 80(7-8), pp.571- 589.
[40] Rothlauf, F., 2006. Representations for genetic and evolutionary algorithms. In Representations for Genetic and Evolutionary Algorithms (pp. 9-32). Springer, Berlin, Heidelberg.
[41] Grosan, C. and Abraham, A., 2007. Hybrid evolutionary algorithms: methodologies, architectures, and reviews. In Hybrid evolutionary algorithms (pp. 1- 17). Springer, Berlin, Heidelberg.
[42] Freitas, A.A., 2003. A survey of evolutionary algorithms for data mining and knowledge discovery. In Advances in evolutionary computing (pp. 819-845). Springer, Berlin, Heidelberg.
[43] Smit, S.K. and Eiben, A.E., 2009, May. Comparing parameter tuning methods for evolutionary algorithms. In 2009 IEEE congress on evolutionary computation (pp. 399-406). IEEE.
[44] Khare, V., Yao, X. and Deb, K., 2003, April. Performance scaling of multiobjective evolutionary algorithms. In International conference on evolutionary multi-criterion optimization (pp. 376-390). Springer, Berlin, Heidelberg.
[45] Yao, X., Liu, Y., Liang, K.H. and Lin, G., 2003. Fast evolutionary algorithms. In Advances in evolutionary computing (pp. 45-94). Springer, Berlin, Heidelberg.
[46] Vikhar, P.A., 2016, December. Evolutionary algorithms: A critical review and its future prospects. In 2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC) (pp. 261- 265). IEEE.
[47] Bartz-Beielstein, T., Branke, J., Mehnen, J. and Mersmann, O., 2014. Evolutionary algorithms. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(3), pp.178-195.
[48] Karafotias, G., Hoogendoorn, M. and Eiben, A.E., 2014. Parameter control in ´ evolutionary algorithms: Trends and challenges. IEEE Transactions on Evolutionary Computation, 19(2), pp.167-187.
[49] item Deb, K., 2011. Multi-objective optimisation using evolutionary algorithms: an introduction. In Multi-objective evolutionary optimisation for product design and manufacturing (pp. 3-34). Springer, London.
[50] Slowik, A. and Kwasnicka, H., 2020. Evolutionary algorithms and their applications to engineering problems. Neural Computing and Applications, 32(16), pp.12363-12379.

[51] Mirjalili, S., 2019. Evolutionary algorithms and neural networks. In Studies in computational intelligence (Vol. 780). Berlin/Heidelberg, Germany: Springer.
[52] E. Cantu-Paz and C. Kamath, ”An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems,” in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 35, no. 5, pp. 915-927, Oct. 2005, doi: 10.1109/TSMCB.2005.847740.
[53] Barros, R.C., Basgalupp, M.P., De Carvalho, A.C. and Freitas, A.A., 2011. A survey of evolutionary algorithms for decision-tree induction. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(3), pp.291-312.
[54] Crepinˇsek, M., Liu, S.H. and Mernik, M., 2013. Exploration and exploitation ˇ in evolutionary algorithms: A survey. ACM computing surveys (CSUR), 45(3), pp.1-33.
[55] Zhang, G., 2011. Quantum-inspired evolutionary algorithms: a survey and empirical study. Journal of Heuristics, 17(3), pp.303-351.
[56] Cheng, R., He, C., Jin, Y. and Yao, X., 2018. Model-based evolutionary algorithms: a short survey. Complex Intelligent Systems, 4(4), pp.283-292.
[57] Derrac, J., Garc´ıa, S., Hui, S., Suganthan, P.N. and Herrera, F., 2014. Analyzing convergence performance of evolutionary algorithms: A statistical approach. Information Sciences, 289, pp.41-58.
[58] Eiben, A.E. and Smit, S.K., 2011. Evolutionary algorithm parameters and methods to tune them. In Autonomous search (pp. 15-36). Springer, Berlin, Heidelberg.
[59] Biethahn, J. and Nissen, V. eds., 2012. Evolutionary algorithms in management applications. Springer Science Business Media.
[60] Veˇcek, N., Mernik, M. and Crepinˇsek, M., 2014. A chess rating system for ˇ evolutionary algorithms: A new method for the comparison and ranking of evolutionary algorithms. Information Sciences, 277, pp.656-679.
[61] Chugh, T., Singh, M., Nagpal, S., Singh, R. and Vatsa, M., 2017. Transfer learning based evolutionary algorithm for composite face sketch recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 117-125).
[62] Oltean, M. and Grosan, C., 2003. A comparison of several linear genetic programming techniques. Complex Systems, 14(4), pp.285-314.
[63] Brameier, M. and Banzhaf, W., 2001. A comparison of linear genetic programming and neural networks in medical data mining. IEEE Transactions on Evolutionary Computation, 5(1), pp.17-26.
[64] Mirjalili, S. and Lewis, A., 2016. The whale optimization algorithm. Advances in engineering software, 95, pp.51-67.
[65] Price, K.V., 2013. Differential evolution. Handbook of Optimization: From Classical to Modern Approach, pp.187-214.
[66] Slowik, A. and Kwasnicka, H., 2020. Evolutionary algorithms and their applications to engineering problems. Neural Computing and Applications, 32, pp.12363-12379.
[67] Deep, K. and Thakur, M., 2007. A new mutation operator for real coded genetic algorithms. Applied mathematics and Computation, 193(1), pp.211-230.