Evolutionary Algorithms: A Critical Review and its Future Prospects

Pradnya A.Vikhar

Research Scholar,
Shri Sant Gadge Baba Amravati University,
Amravati, India
pradnyav123@gmail.com

Abstract: **Evolutionary algorithm (EA) emerges as an important optimization and search technique in the last decade. EA is a subset of Evolutionary Computations (EC) and belongs to set of modern heuristics based search method. Due to flexible nature and robust behavior inherited from Evolutionary Computation, it becomes efficient means of problem solving method for widely used global optimization problems. It can be used successfully in many applications of high complexity.**

This paper presents a critical overview of Evolutionary algorithms and its generic procedure for implementation. It further discusses the various practical advantages using evolutionary algorithms over classical methods of optimization. It also includes unusual study of various invariants of EA like Genetic Programming (GP), Genetic Algorithm (GA), Evolutionary Programming (EP) and Evolution Strategies (ES). Extensions of EAs in the form of Memetic algorithms (MA) and distributed EA are also discussed. Further the paper focuses on various refinements done in area of EA to solve real life problems.

Keywords: Evolutionary Computations, Evolutionary Algorithm, Memetic Algorithms, Distributed EAs

I. INTRODUCTION

Now days, the complexity of real world application is increased substantially. The problems like robotics, operation research, decision making, bioinformatics, machine learning, data mining and many more are very complex and hard to solve[1][2]. An approach suggested to tackle such complex problems inspired by Darwinian natural evolution is referred as Evolutionary Computations. Evolutionary computing involves various algorithms, commonly known as Evolutionary Algorithms (EAs) [2][3]. EA uses simulated evolution to explore the solutions for complex real world problems. Evolutionary process is best suited to the applications where it is not possible to use heuristic solutions and may lead to inadequate results. EA is receiving remarkable interest, particularly with the way; it is applied for solving the practical problem. During last two decades Evolutionary Algorithms becomes very popular tool for searching, optimization and providing solutions to complex problems [4]. EA is based on the principle of survival of the fittest (i.e. evolution). It is highly inspired by Darwinian evolutionary [5] concept that changes the system incrementally over time. It works on some basic principles like; one or more populations of individuals are present and they are competing for limited resources. The populations changes dynamically and it will always search the space of possible forms (the fitness landscape) for the individual that are best adapted.

In nature, individuals need to adapt to their environment in order to survive, this process is known as evolution. At the time of reproduction, the features which makes individual more suited to compete are preserved and weaker elements are eliminated. Genes are the units which control these features, set of such genes forms chromosomes. Only the fittest individuals survive in the consequent generations and their fittest genes are transmitted to their descendants during the process of recombination. It is called as crossover [6][7]. This process of natural selection and optimization leads to the growth of "evolutionary algorithms" [8].

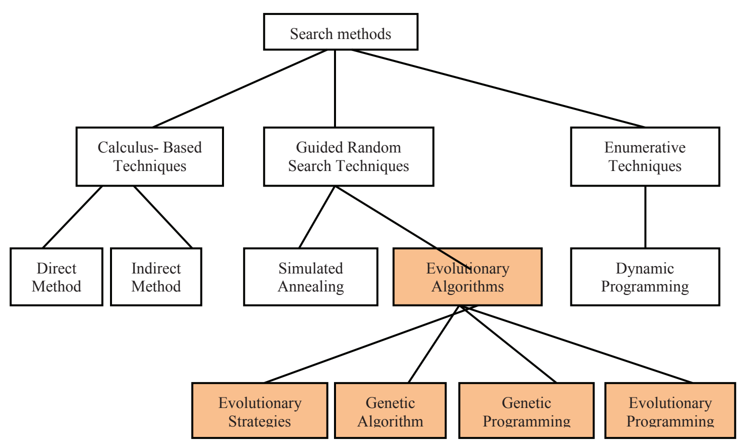

Fig.1 Searching Methods

II. EVOLUTIONARY ALGORITHM

Evolutionary algorithm (EA) is a subclass of Evolutionary computation and belongs to set of general stochastic search algorithm. Figure 1 shows the place of EA in different searching methods. It is a metaheuristc optimization algorithm worked on concept of population [3]. Metaheuristics are the higher-level procedures intended to find, produce, or select a lower- level procedures or heuristics which may performs partial search. It is applicable to various optimization problems with limited computation capability and having insufficient or imperfect information. In such situations, it provides adequately good solution.

An EA is inspired by the mechanism of biological evolution, such as reproduction, mutation, recombination, and selection. A set of candidate solution (i.e. elements of function domain) is created randomly to maximize the quality function. Then quality function in the form of abstract fitness function is applied to problem domain. For next generation some better candidates are selected on the basis of fitness function. This is achieved by applying the technique of recombination and/or mutation to them. Recombination is represented by the binary operator. This operator can be applied to two or more selected candidates known as parents and it generates one or more new candidates (children) as a result. Whereas mutation is applied to only one candidate and it results in one new child. After executing this recombination or mutation, it generates a set of new candidates on the basis of their fitness function. This is an iterative process. It can be continued until sufficiently good quality candidate is found [1][2]. In the following section the diagrammatic view (figure 1) of simple EA is shown and it is thoroughly explained along with terminology used in EA[2] [9].

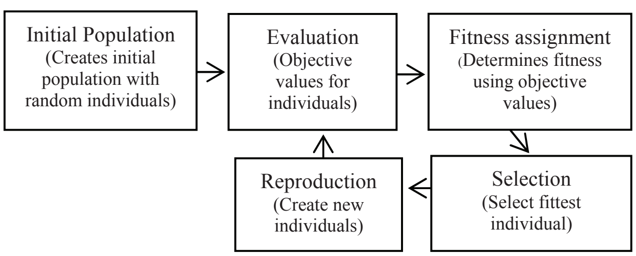

Fig.2 A generic view for simple Evolutionary Algorithm

A. Steps involved in solving the problem with EA [2][9]

Step 1: In the first step original problem space and problem solving space is defined. It is known as representation. It is the space where the evolution takes place. The purpose of representation is to bridge the gap between real world problems and EA world. In the original problem space the objects which form possible solutions are referred as phenotypes. Their corresponding encoding, the individuals within EA is referred as genotypes.

Step 2: It involves determining evaluation Function (Fitness Function). This function is used as the basis for process of selection, and it enables improvements.

Step 3: Once defining representation population holds possible solution. It is a multistep of genotype which forms the unit of evolution. For given representation, population is number of individuals in it.

Step 4: The individuals are determined depend on their quality by parent selection. Due to this, good quality individuals become parent in next generation. If an individual is selected as a parent, it undergoes variation to create offspring.

Step 5: Variation operators create new operators from the old one. There are two types of variation operators namely mutation and recombination. Mutation is represented by unary operator. When it is applied to one genotype, it generates the child of it. Whereas recombination is a binary operator. Information of two or more parent genotypes is combined during the process of recombination and it results into one or more offspring genotypes.

Step 6: Survivor selection distinguishes individuals on the basis of their quality. It is quite similar to process of selection of parent. But it takes place in different stages of the evolution. This mechanism is ensued only if there are offspring of selected parents.

Step 7: The first population is generated by randomly generated individuals. Usually, specific heuristic with higher fitness is used to form an initial population. For termination condition, as problem knows optimum fitness level, then reaching this level can stop process. But being stochastic, EA does not guarantee to reach optimal solution. Therefore it never meets to optimal fitness value and algorithm never stops. So it needs a condition which certainly stops the algorithm. Some conditions used for the same are maximum CPU time, total number of evaluations, given period of time etc.

III. TYPES OF EAs

Sub area of Evolutionary algorithm includes Evolutionary Programming (EP), Evolution Strategies (ES), Genetic Programming (GP) and Genetic Algorithm (GA) [6][8][9][10]. All EAs work on common principle of simulated evolution of individual using the process of selection, mutation and reproduction. But one can differentiate them on the basis of their implementation and the way they applied to particular problem.

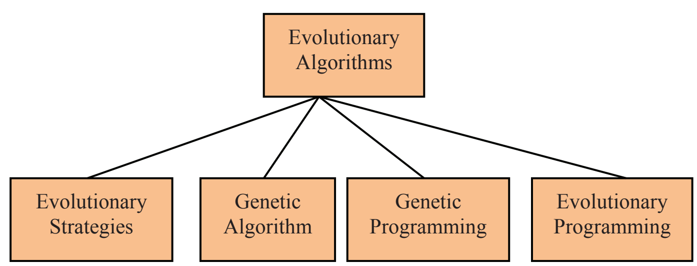

Fig 3. Classification of Evolutionary Algorithms

A. Genetic algorithm (GA):

The most popular type of EA is Genetic Algorithm (GA) as it exhibits the clearest mapping of natural evolution process onto to computer. It often used [11] for machine learning, pattern recognition and optimization problem. It was first proposed by Holland and his student in 1970, primarily used for adaptive search and adaptive system design. GA uses recombination and mutation operators to seek the solution of a problem. The solution is in the form of strings of numbers (traditionally binary) i.e. a Bit-String representing the genes.

B. Genetic programming

It was developed by Koza in 1992. Unlike GA in which attributes are represented using general binary coding, Genetic programming represents programs or instruction sets as attributes. Thus GP generates the solutions in the form of computer programs, and its ability to solve a computational problem is determined by their fitness function. Due to trees based encoding genetic programming , applicable to areas [11] like Arithmetic operations , Mathematical functions, Boolean Operations (e.g., AND, OR, NOT), Recursive Functions etc.

C. Evolution strategy

It was first proposed in 1973 by Rechenberg as an optimization method for complex, multimodal and non-differentiable function. It is used to solve the actual expression of an attribute by omitting any redundant coding. Thus in evolution strategy the solutions are represented using vectors of real numbers and it uses self-adaptive mutation rates. Some well-known applications of evolution strategies are [11] routing and networking, Biochemistry, Optics, Engineering design.

D. Evolutionary programming

The concept of EP was first suggested by Fogel in 1966 as a method to achieve artificial intelligence. Its working is quite closed to working of Evolutionary Strategies. Unlike ES, it has no restriction regarding the use of data types of attributes. EP has fixed structure of program and it allows numerical parameters to evolve. Forecasting, Generalization, Games and Automatic control are some suitable areas of application of evolutionary programming [11].

Which flavors of EAs is good to used, it has been associated with different representations. Best way to select particular strategy for solve a problem is to select proper representation suitable to the problem and then select variation operators to suit the representation. Selection operators are independent of representation because they only use fitness function.

IV. ADVANTAGES OF EVOLUTIONARY ALGORITHM

EAs have been widely used in many complex problem solving applications, due to its many advantages over classical search and optimization techniques [3][6][9].

- As it is inspired by natural evolution, it is conceptually simple and flexible.
- It utilizes prior information. It is obvious that the method that considers prior information about problem will outperform a method using less information and it will also restrict the search space.
- Some numeric techniques available are only applicable for applications having continuous values or other having constrained sets. But EA is representation independent.
- Evolution is a parallel process. Each evaluation in EA performs parallel operations and only operations performed during selection process requires some serial processing.
- Optimization using traditional methods changes with dynamic variation occurred in problem environment. It is not robust. It always requires a complete restart to provide a solution. Whereas evolutionary algorithms are robust and develop to adapt solution in changing environment.
- EA proves its ability to solve problems without any human expertise. However it uses human expertise if it is available, but human expertise may not be self-consistent or qualified or may be in error. So it does not result satisfactorily for automating problem solving routines.

Besides of many advantages EA suffers from some problems like it does not assured to always give an optimal solution to specific problem within predictable time, it may need parameter tuning by trial-and-error, it needs for lots of computational resources.

V. EXTENSIONS TO EAs

To increase overall performance of EA methods two extensions are suggested namely Memetic Algorithms (MA) and distributed EA [12][13].

A. Memetic algorithms [12]

The heuristics which adopts an individual are known as local search. Memetic algorithm (MA) is a combination of the local search with the EA. In nature, in order to survive the individuals changes their properties after birth and tries to adapt in the surroundings. This process of adaption is called plasticity. In EA every individual can easily get such plasticity and the way individuals adopted are selected arbitrarily. The weight between the local search and the EA method can be shifted randomly. In this way the EA is extended to MA, will perform good multi-start search of the local search heuristic [12].

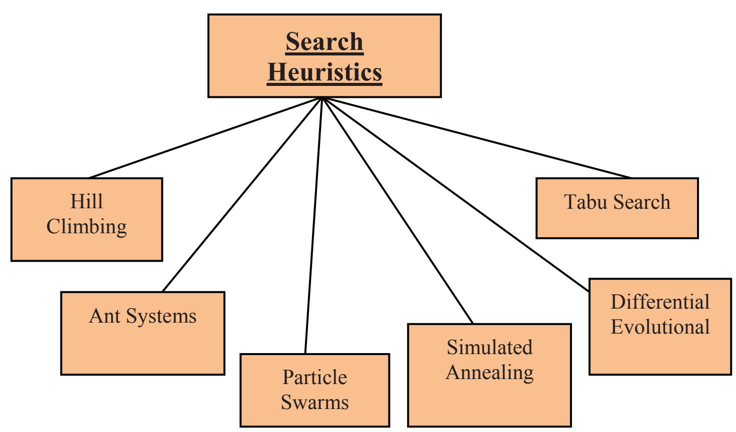

Fig 4. Memetic Algorithms

B. Distributed Evolutionary Algorithms [6][12][13]

In execution of EA methods large amount of computational resources are required. There are two basic reasons for this. First possibility is that there may be very slow evaluation of the fitness functions. Secondly, size of population may become too large. To tackle this problem the obvious solution is to distribute the whole work among several computers and performs all computations in parallel. In nature, species have a custom to specialize, to explore and to grow into new species which never found anywhere. EA also exchanges best available individual from time to time and all sub populations have inclination to meet the best solution of all sub optima. In this way, distributed EAs use parallelism which highly reduces the computational time. It also increases the quality of the solution found in multimodular search spaces [12][13].

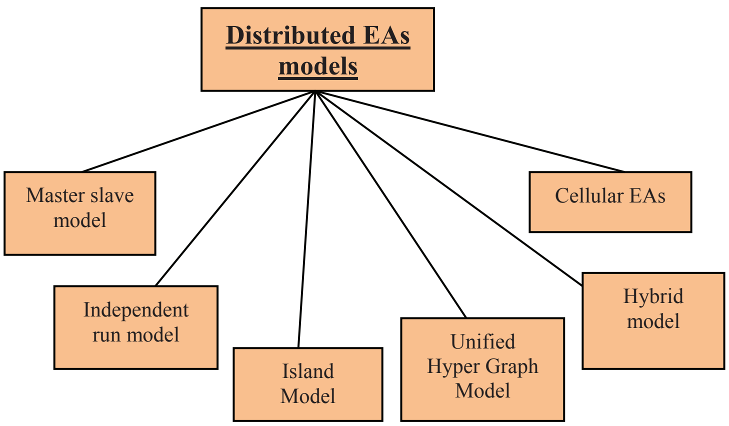

Fig 5. Distributed Evolutionary Algorithms

VI. FURTHER REFINEMENTS IN EVOLUTIONARY ALGORITHMS

Though Evolutionary Algorithms are widely applicable to many domains, it delivers only marginal performance. Hence the present efforts in area of EA are focused to apply some complementary algorithms. It may enhance the performance of EA [12]. A current trend is to hybridize two or more algorithm or to improve the existing algorithms. It will obviously give better results than individual. The following table shows the some recent hybridized as well as improved/ refined EA with their description-

TABLE1 REFINEMENTS IN EA

Sr.No	EA Technique	Description
1	Genetic Swarm Optimization (GSO) [14]	Combines GA and PSO to solve electromagnetic problem
2	Hybrid PSO [14]	It reduces probability of trapping local optima using Cauchy mutation for PSO
3	Self-adaptive differential evolution (SaDE) [15]	It adaptively search for suitable strategy and associated parameter setting
4	Immune self-adaptive differential evolution (ISDE) [15]	Scale and crossover factors of DE are adaptively modified by information process mechanism of biological immune system
5	Multi-objective Evolutionary algorithm (MOEAs) [16] [19]	Extension to simple evolutionary multi-objective optimizer
6	Multi-objective Evolutionary algorithm based on decomposition (MOEA/D) [17]	Based on simple evolutionary multi-objective optimizer and decomposition
7	Dynamic multi-agent GA [17]	Integrates dynamic multi-agent with Genetic algorithm
8	Multi-objective Particle Swarm Optimization (MOPSO) [18]	Given multi-objective functions to solve optimization problem

VII. CONLUSION

The complete overview of Evolutionary algorithm is presented in this paper. Due to simplicity, flexibility and robustness, EA becomes popular problem solving technique in learning and optimization. With many sub types Evolutionary Algorithm leads to be one of the promising areas to solve wide range of problems. Best suited sub EAs are selected based on representation of problem. GA and EP are most popular approaches used for binary representation. ESs is used for real number parameter optimization whereas functions or computer programs are optimized using GP. Further EAs are extended to MA and distributed EA to increase the search performance. Different approaches of EAs are combined which generates many refinements to solve variety of real life applications. Due to extension and possible refinement, EA achieves remarkable attention from many researchers now days.

REFERENCES

[1] Kenneth A. DeJong, Evolutionary Computation: A unified Approach, the MIT Press, Cambridge, Massachusetts, Landon, England., 2006

[2] A.E.Eiben and J.E. Smith, “Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Hedelberg, Germany, 2003.

[3] Nada M.A. AL-Salami, “Evolutionary Algorithm Definition”, American J. of Engineering and Applied Sciences, Vol 2, Issue. 4,pp.789-795, 2009.

[4] Xin Yao, “Global Optimization by Evolutionary Algorithms” IEEE Transactions on Evolutionary Computation,Vol:3 , Issue: 2, pp.82 -102 , Jul. 1999.

[5] Karthik Sindhya,“An Introduction to Evolutionary Algorithms”, Industrial Optimization Group, Department of Mathematical Information Technology, University of Tyvaskyla.

[6] Darrell Whitley, “An Overview of Evolutionary Algorithms: Practical Issues and Common Pitfalls”, Elsevier, Information and Software Technology, Vol. 43 , Issue. 14, pp. 817-831, Nov. 2001.

[7] Zbigniew Michalewicz, Robert Hinterdingy ,Maciej Michalewiczz, “Evolutionary Algorithms”, Fuzzy Evolutionary Computation, pp 3-31,1997,

[8] Available: http://neo.lcc.uma.es/opticomm/introea.html

[9] Pradnya Vikhar, “Evolutionary Algorithm: A Classical Search and Optimization Technique”, International Journal of Pure and Applied Research in Engineering and Technology, Volume 4, issue 9, pp. 758-766, 2016

[10] Kenneth De Jong, David B Fogel and Hans-Paul Schwefel, "A history of evolutionary computation", 1997 IOP Publishing Ltd and Oxford University Press Handbook of Evolutionary Computation release 97/1.

[11] Carlos A. Coello Coello, 'An Introduction to Evolutionary Algorithms and Their Applications" Advanced Distributed Systems Lecture Notes in Computer Science Volume 3563, 2005, pp 425-442, 2005.

[12] Felix Streichert, "Introduction to Evolutionary Algorithms", Ph D, Thesis University of Tuebingen,

[13] Dirk Sudholt, "Parallel Evolutionary Algorithms", Chapter in the Handbook of Computational Intelligence, University of Sheffield, UK, 2007

[14] Sarvesh S. Cahkradeo, Aishwarya Hendre, Shantanu Deshpande, "Generalized Theory for Hybridization of Evolutionary Algorithms", IEEE International conference on Computational Intelligence and Computing Research, pp. 1-5, 2014

[15] Chengfu Sun, Haiyan Zhou, Liqing Chen, "Improved differential Evolutional Algorithms", IEEE International Conference on Computer Science and Automation Engineering, pp. 142-146, 2012

[16] Yuan-Long Li, Yu-Ren Zhou, Zhi-Hui Zhan, Jun Zhang, "A primary Theoretical Study on Decomposition Based Multiobjective Evolutionary Algorithm", IEEE Transactions on Evolutionary Computation, Volume: 20, Issue: 4, pp. 563 – 576, 2015

[17] Jing Liu, Yaxiong Chi, Chen Zhu, "A dynamic Multi-agent Genetic Algorithm for Gene Regulatory Network Reconstruction based on Fuzzy Cognitive Maps", IEEE Transactions on Fuzzy Systems, Volume: 24, Issue: 2, pp. 419 – 431, 2015

[18] Yujiao Zeng, Yanguang Sun, "Comparison of Multiobjective Particle Swarm Optimization and Evolutionary Algorithms for Optimal Reactive Power Dispatch Problem", IEEE Congress on Evolutionary Computation (CEC), pp. 6-11, 2014

[19] Jun He, Boris Mitavskiy, Yuren Zhou," A theoretical Assessment of Solution Quality in Evolutionary Algorithms for the Knappsack Problem", IEEE Congress on Evolutionary Computation (CEC), 2014