ISSN : 2231-0711
Kunjal Bharatkumar Mankad | IJCSET(www.ijcset.net) | August 2015 | Vol 5, Issue 8,304-308

A Comparative Analysis of Design Issues of Evolutionary Structures

Kunjal Bharatkumar Mankad
Rajkot, India

Abstract-**Evolutionary computing is ubiquitous in nature. Adaptive nature and evolutionary search capabilities of evolutionary structures produce a more efficient exploration of the state space of possible solutions. Evolutionary computing provides four major structures namely evolutionary programming, genetic programming, genetic algorithm and evolutionary strategy. The uniqueness of evolutionary structure is that they are capable to provide solutions for highly complex mathematical problems very efficiently. The paper explains differences between traditional search and optimization algorithm and evolutionary algorithm along with advantages of evolutionary computing. Evolutionary life cycle and procedural characteristics of each structure are discussed in details. Structural parameters of evolutionary methods such as chromosomal representation, encoding, selection, crossover and mutation are narrated comparatively.The paper concludes by showing justification of design issues of evolutionary structures.**

Keywords: Evolutionary Algorithm, Evolutionary Computing, Evolutionary Programming, Evolutionary Strategy, Genetic Algorithm, Genetic Programming.

I. INTRODUCTION

Evolutionary Computation (EC) refers to the computer-based problem solving systems that use computational models of evolutionary process.Evolutionary algorithms have provided numerous advantages compared to conventional methods. The capability to provide simulation of biological evolution leads evolutionary methods to design highly complex mathematical problems. Evolutionary algorithms provide adaptive nature with multimodal characteristics which proves quite favorable in design of evolutionary systems. EA are successfully applied to varieties of numerous problems from different domains, including intelligent search, optimization, automatic programming, machine learning, operations research, bioinformatics, and social systems, engineering systems, medical diagnostic, stock and share market, forecasting systems and many more.

The introductory section of the paper explains importance of evolutionary algorithms. The second section of the paper explains the basic procedure of evolutionary algorithm. Differences between traditional search and optimization algorithm and evolutionary algorithm are presented in this section. It also describes advantages of evolutionary algorithm. The third section explains evolutionary structures such as evolutionary programming, genetic algorithm, genetic programming and Evolutionary Strategies in detail.Evolutionary life cycle and procedural explanation of each evolutionary structure is presented in this section. The forth section of the paper presents comparison among evolutionary characteristics such as representation, encoding, and types of genetic operators. The concluding part of the paper justifies the design perspective of evolutionary structures.

II. EVOLUTIONARY ALGORITHMS

In recent years, cognitive systems have gained prominence by implementing evolutionary approach to the computational modeling. Computational paradigm which is based on evolutionary characteristics is popularly known as evolutionary computing. Evolutionary computing is based on principles of natural genetics. It is basically the type of computing which structuresthe problem with the help of process of evolution. The process of evolution is the change in the inherited traits of a population from one generation to the next. Evolutionary Computation (EC) refers to the computer-based problem solving systems that use computational models of evolutionary process. EC uses an algorithm to implement evolutionary characteristics in solving problems. Such algorithm is known as evolutionary algorithm. Evolutionary characteristics are achieved using evolutionary computing in which problem is structured using chromosome of different types in form of symbols or alphanumeric numbers. Evolutionary algorithms are basically computational models of evolutionary processes. EA are successfullyapplied to varieties of numerous problems from different domains, including intelligent search, optimization, automatic programming, machine learning, operations research, bioinformatics, and social systems, engineering systems, health care, medical diagnostic, stock and share market, forecasting systems and many more. Fig. 2 represents basic procedure of evolutionary computing [1].

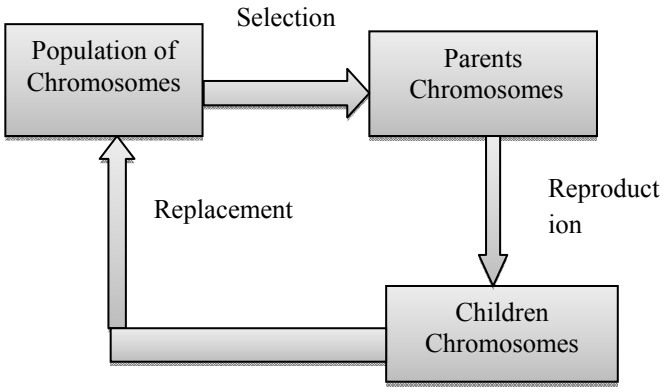

Fig.1: Basic Elements of Evolutionary Computing

A. *Difference of Traditional optimization Algorithms Vs Evolutionary Algorithms*

There are significant differences observed between GA and most of the traditional optimization algorithms as summarized by [2,3,4,5,6]:

- Evolutionary algorithms are ubiquitous in nature while traditional algorithms are static in nature.
- Evolutionary algorithm are specially designed to solve mathematical function,which describes the problem is not known and the values at certain parametersare obtained from simulations.
- Evolutionary algorithm can handle multi-modal functions whereas many other optimizationtechniques supports only single model function;
- In traditional optimization, convergence to an optimal solution depends on the chosen initial solution while in EA, due to randomness , initial solution is always different;
- EA is generic in nature due to objective function while a traditional algorithm is efficient in solving one problem but the same may not be efficient in solving a different problem;
- EA works with coding of parameter set rather than actual value of parameters;
- A traditional algorithm may not be efficient to handle problems with discrete variables or highly non-linear variables with constraints while EA can be robustly applied to problems with any kinds of objective functions, such as nonlinear or step functions; because only values of the objective function for optimization are used to select genes;
- Traditional algorithm can stuck at suboptimal solutions while EA can have less chance to be trapped by local optima due to characteristics of crossover and mutation operators.

B. *Advantages of Evolutionary Algorithms*

EA offers following advantages [6,8]:

- A primary advantage of evolutionary computation is that it is conceptually simple. The procedure may be written as difference equation (1)
  $x[t + 1] = s(v(x[t]))$ (1)
- wherex[t] is the population at time t under a representation x, v is a random variation operator, and sis the selection operator .
- Evolutionary algorithm provides representation independent performance, in contrast with other numerical techniques, which might be applicable for only continuous values or other constrained sets.
- Evolutionary algorithms offer a framework such that it is comparably easy to incorporate prior knowledge about the problem. Incorporating such information focuses the evolutionary search, yielding a more efficient exploration of the state space of possible solutions.
- Evolutionary algorithms can also be hybridized with other techniques. For ex. This may be as simple as the use of a gradient minimization used after primary search with an evolutionary algorithm (for example fine tuning of weights of a evolutionary neural network), or it may involve simultaneous application of other algorithms (e.g., hybridizing with simulated annealing or tabu search to improve the efficiency of basic evolutionary search).
- The evaluation of each solution can be handled in parallel and only selectionrequires some serial processing. Implicit parallelism is not possible in many global optimization algorithms like simulated annealing and Tabu search.
- Traditional methods of optimization are not robust to dynamic changes inproblem the environment and often require a complete restart in order to provide a solution (e.g., dynamic programming). In contrast, evolutionary algorithms can be used to adapt solutions to changing circumstance.
- EA are capable to handle problems which replaces human expertise although human expertise should be used when it is available, it oftenproves advantageous for automating problem-solving routines.

III. EVOLUTIONARY STRUCTURES

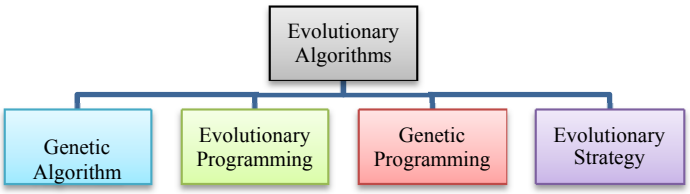

Fig. 2: Evolutionary Structures

A. *Evolutionary Algorithm*

A technique borrowed from the theory of biological evolution that is used to create optimization procedures or methodologies. In nature, evolution is mostly determined by natural selection or different individuals competing for resources in the environment. Those individuals that are better are more likely to survive and propagate their genetic material.EAs fall into the category of "generate and test" algorithms.They are stochastic,population-based algorithms. EA are based on variation operators (recombination and mutation) to create the necessary diversity and finally introduce novelty. EA also use selection operator whichreduces diversity and provides good quality solution.Fig.3 represents pseudo code of an evolutionary algorithm [7].

```
BEGIN
INITIALISE population with random candidate solutions;
EVALUATE each candidate;
REPEAT UNTIL (TERMINATION CONDITION is satisfied)
DO
SELECT parent;
RECOMBINE pairs of parents;
MUTATE the resulting offspring;
SELECT individuals for the next generation;
END
```

Fig.3: Pseudo Code of Evolutionary Algorithm

B. Evolutionary Programming (EP)

The evolutionary programming (EP) paradigm concentrated on models involving a fixed-size population of N parents, each of which produced a single offspring. The next generation of N parents was determined by combining both parents and children into a single population of size 2N, rank ordering them by fitness and allowing only the top N to survive [8]. Evolutionary Programming (EP) is originated from Fogel [9,10]. It is a useful method of optimization when other techniques such as gradient descent or direct, analytical discovery are not possible to implement. It is capable to predict a symbol based on previous symbol. In EP, the chromosomes are presented as Finite State Machines. A finite state machine provides machine possessing with a finite number of different internal states. The basic EP method involves 3 steps (Repeat until a threshold for iteration is exceeded or an adequate solution is obtained) [11]:

- Choose an initial population of trial solutions at random. The number of solutions in a population is highly relevant to the speed of optimization. But there is not any specification possible regarding number of appropriate solutions and inappropriate solutions.
- Each solution is replicated into a new population. Each of these offspring solutions are mutated according to a distribution of types of mutation ranging from minor to extreme.

Each offspring solution is assessed by computing its fitness values. Typically, a stochastic tournament is held to determine N solutions to be retained for the population of solutions, although this is occasionally performed deterministically.The size of the population and the intensity of mutation are different for each different application.

C. Genetic Programming (GP)

Genetic programming is the extension of evolutionary learning into the space of computer programs. Genetic Programming (GP) technique provides a framework for automatically creating a working computer program from a high-level problem statement of the problem [6].

Genetic programming creates computer programs in the LISP or scheme computer languages as the solution as LISP provides the same structure for data as well as programs [12]. This type of procedure can make manipulation and evaluation of population very easy [6].Specifically, genetic programming iteratively transforms a population of computer programs into a new generation of programs by applying analogs of naturally occurring genetic operations. The chromosome in a GP systems form hierarchical rather than linear structure. The basic procedure as under [6]:

- Generate an initial population of random compositions of the functions and terminals of the problem (computer programs);
- Compute the fitness values of each individual in the population ;
- Apply some selection strategy and suitable reproduction operators produce two offspring;
- Procedure is iterated until the required solution is found or the termination conditions have reached (specified number of generations).

The above operations are applied to computer programs in the population selected with a probability based on fitness. In Genetic Programming, the crossover operator is essential to introduce diversity of the population. Mutation operation is applied before crossover. The flavors of Genetic Programming are as under [6]:

- Linear genetic programming (LGP)
- Gene expression programming (GEP)
- Multi-expression programming (MEP)
- Cartesian genetic programming (CGP)
- Traceless genetic programming (TGP)
- Gramatical evolution (GE)
- Genetic algorithm for deriving software (GADS).

D. Genetic Algorithm

Genetic Algorithms are the most popular extension to Genetic Programming. The area of GA has been traversed by three prominent researchers namely Fraser in 1962, Bremermann in 1962 and Holland in 1975 [13,14,15]. Genetic Algorithms are pioneered by John Holland in 1970's [16]. Genetic Algorithms are based on principle of natural evolution which is popularly known as "Darwinian Evolution". Genetic Algorithm is an evolutionary-based search or optimization techniques that perform parallel, stochastic, but direct search method based on natural genetics to evolve the best solution. Apart from search characteristics, GA provides quality of optimization, hybridization and parallel processing. Engineering, scientific as well as business applications demand GA very widely. The solutions provided by Genetic Algorithms are very rapid, reliable and precise. Fig.4 represents basic components of GA [17]. The basic structure of simple genetic algorithm is as follows [18]:

1: Generate random population of n chromosomes.
2: Evaluate the fitness value of each rule set using fitness function and a set of test instances.
3: Create a new population by repeating following operators until the new population is complete;
   a. Selection, b. Crossover & c. Mutation.
4: If the termination criterion has been reached, then return the best solution from current solution otherwise go to step 2.

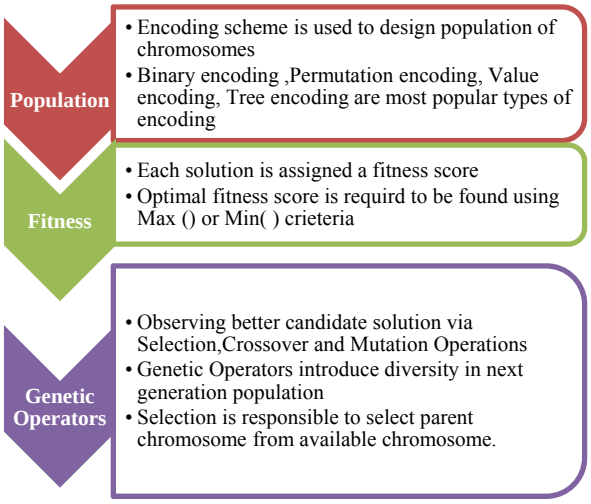

Fig. 4: The Basic Components of Simple Genetic Algorithm

E. Evolutionary Strategies (ES)

Evolution Strategy (ES) was developed by Rechenberg [19] at Technical University,Berlin. ES is developed for the problems which are difficult to problem mathematically. The system to be optimized is actually constructedand ES is used to find the optimal parameter settings. Evolution strategiesmerely concentrate on translating the fundamental mechanisms of biologicalevolution for technical optimization problems. ES works on two types of parameters using vector of real number representation [6].

1. Object Parameters- op
2. Strategy parameters- sp

Both of these parameters design the data structure for a single element. An example of ES can be represented as follows:

A population P ofn individuals could be described as P = (c1, c2, . . . , cn−1, cn), where the ithchromosome ci is defined as ci = (op, sp) with op = (o1, o2, ..., on−1, on) andsp = (s1, s2, ..., sn−1, sn).

Evolution strategies derive inspiration from principles of biological evolution. It is based on following two rules of evolution [1]:

1. It randomly changes all variables at a time ,and
2. If the new set of variables does not deteriorate the fitness keep it, else discard.

Initially, the strategy based on above stated rules was popularly known as simplest ES which became later the popular (1+1) ES. Evolution strategies typically use real-valued vector representations. Initially they were only working with the help of mutation operation. Evolution strategies are primarily employed for continuous parameter optimization. In a generational procedure, following steps are executed [20]:

A. One or several parents are picked from the population (mating selection) and new offspring are generated by duplication and recombination of these parents;
B. The new offspring undergo mutation and become new members of the population;
C. Environmental selection reduces the population to its original size.

IV. Comparison Among Members Of Evolutionary Computing

All evolutionary structures share a common conceptual base of simulating the process of evolution of individual structures via processes of selection, mutation, and reproduction. The processes depend on the perceived performance of the individual structures as defined by the problem [21]. Procedural parameters such as encoding, chromosomal representation, adaptive-ness, fitness, selection, crossover and mutation are compared as presented in Table 1.

V. Conclusion

The paper has provided an extensive review of major evolutionary methods to design evolutionary systems. In the area of evolutionary computing, there are four different structures developed to design evolutionary process. These include evolutionary programming, genetic programming, genetic algorithm and evolutionary strategy. The major contribution of these methods is to provide efficient Search and optimization for complex mathematical problems. Evolutionary algorithms are more advantageous in many aspects compared to traditional search and optimization methods. The paper has explained significance and working characteristics of all major evolutionary methods. The procedure to develop an algorithm their design requirements are discussed in detail. Different characteristics such as representation, encoding, types of crossover and mutation operationsof each of evolutionary methods are identified and compared with one another.

TABLE 1
COMPARISON AMONG CHARACTERISTICS OF EVOLUTIONARY STRUCTURES

Characteristics	EP	GP	GA	ES
Encoding	Real Value	Expression	Binary/Permutation/ Value/Tree	Real Value
Chromosomal Representations	Finite State Machines	LISP Trees	Binary Strings	Real Vectors
Adaptive- ness Measure	Variance	Not Possible	Not Possible	Standard Deviation, covariance
Fitness	Scaled objective function value	Scaled objective function value	Scaled objective function value	Objective Function Value
Selection	Probabilistic	Probabilistic	Probabilistic	Deterministic
Recombination/Crossover	None	Main Operator	Main operator	Different types for self-adaption
Mutation	Only Operator	Background	Background	Main Operator

REFERENCES

1. Mankad,K.B., An Architectural Perspective of Soft Computing Methods, International Journal of Emerging Research in Management and Technology (IJERMT),Vol.4, Issue 1, February 2015
2. Karry, F. O. and Silva, C. D. Soft computing and intelligent system design: Theory, tools and applications, 1st ed., New York, NY: Pearson, 2004
3. Padhy, N. P., Artificial Intelligence and Intelligent System. New Delhi, India: Oxford University Press, 2005
4. Rajsekaran, S. and Pai, V., Neural Networks, Fuzzy Logic, and Genetic Algorithms Synthesis and Applications. New Delhi: PHI, 2003.
5. Deb, K. Revolutionary Optimization by Evolutionary Principles, [online]Available: http://www.iitk.ac.in/directions/directsept04/deb~new.pdf
6. Abraham et al.: Evolutionary Computation: from Genetic Algorithms to Genetic Programming, Studies in Computational Intelligence (SCI) **13**, 1–20 (2006)
7. A.E.Eiben and J.F.Smith, 2003, Introduction to Evolutionary Computing, 1st edition,ISBN:3-540-40184-9 [Online] Available at www2.cs.uh.edu/~ceick/6367/eiben2.ppt
8. Dejong, KA. Evolutionary Computation A unified approach, The MIT Press, Cambridge,2006
9. L. J. Fogel,"Autonomous Automata", Industrial Research, Vol. 4.
10. D.B.Fogel, System Identification through Simulated Evolution: A Machine Learning approach to modelling, Ginn Press,1991.
11. Lecture notes [Online].Available at :http://www.cs.cmu.edu/Groups/AI/html/faqs/ai/genetic/part2/faq-doc-3.html
12. History of Lisp, http://www-formal.stanford.edu/jmc/history/lisp.html
13. Fraser, A.S., Simulation of genetic systems, J. Theoretical Biology, vol. 2, no.3, pp. 329- 346, May 1962.
14. Bremermann, H. J., Optimization through evolution and recombination, in Self –organizing Syst., M.C. Yovits, et al., Eds. Washington D.C.: Spartan Books, 1962, pp. 93-106.
15. Holland, J. H., Adaptation in natural and artificial systems. Ann arbor: The University of Michigan Press, 1975.
16. Holland, J. H., Hierarchical descriptions of universal spaces and adaptive systems, in Essays on cellular automata, A.W. Bruks , Ed. Urbana: Univ. Illinois Press,1970, pp. 320-353.
17. Mankad, K. B., 2014. The Significance of Genetic Algorithms in Search, Evolution, Optimization and Hybridization: A Short Review. International Journal of Computer Science and Business Informatics, Vol. 9, No. 1, pp. 103-115. J.R.Koza. Genetic Programming II: Automatic Discovery of Reusable Programs.MA: MIT Press,1994
18. Herrera, F. (1997). Ten Lectures on Genetic Fuzzy Systems, Technical report ,SCCH-TR-0021, Spain.
19. Rechenberg, I., Evolutionsstrategie: OptimierungtechnischerSystemenachPrinzipien der biologischen Evolution, Stuttgart: Fromman-Holzboog, 1973.
20. N. Hansen, D. V. Arnold and A. Auger, Evolution Strategies,April, 2013.
21. G. Barış, Evolutionary algorithms, available at [Online] Available at: http://robot.cmpe.boun.edu.tr/robsem/evolutionaryAlgorithms.ppt Retrieved on 15 June 2015