Hybrid Nature-Inspired Algorithms: Methodologies, Architecture, and Reviews

Abhishek Dixit, Sushil Kumar, Millie Pant and Rohit Bansal

Abstract Evolutionary computation has turned into a significant problem-solving approach among several researchers. As compared to other existing techniques of global optimization, the population-based combined learning procedure, robustness, and self-adaptation are some of the vital topographies of evolutionary algorithms. In spite of evolutionary algorithms has been broadly acknowledged for resolving numerous significant real applications in various areas; however in practice, occasionally they carry only fringe performance. There is slight motivation to assume that one can discover an unvaryingly finest optimization algorithm for resolving all optimization problems. Evolutionary algorithm depiction is resolute by the manipulation and survey liaison retained during the course. All this evidently elucidates the necessity for fusion of evolutionary methodologies, and the aim is to enhance the performance of direct evolutionary approach. Fusion of evolutionary algorithms in recent times is gaining popularity owing to their proficiencies to resolve numerous legitimate problems such as, boisterous environment, fuzziness, vagueness, complexity, and uncertainty. In this paper, first we highlight the necessity for fusion of evolutionary algorithms and then we explain the several potentials of an evolutionary algorithm hybridization and also discuss the general architecture of evolutionary algorithm's fusion that has progressed all through the recent years.

A. Dixit (✉) · S. Kumar
Amity University, Noida, India
e-mail: abhishekdixitg@gmail.com

S. Kumar
e-mail: kumarsushiliitr@gmail.com; skumar21@amity.edu

M. Pant
Indian Institute of Technology, Roorkee, Uttarakhand, India
e-mail: millifpt@iitr.ac.in

R. Bansal
Rajiv Gandhi Institute of Petroleum Technology, Noida, India
e-mail: rohitbansaliitr@gmail.com

M.S. Reddy et al. (eds.), *International Proceedings on Advances in Soft Computing, Intelligent Systems and Applications*, Advances in Intelligent Systems and Computing 628, https://doi.org/10.1007/978-981-10-5272-9_29

Keywords Nature inspired computation (NIC) · Evolutionary computation (EC) Genetic algorithm (GA) · Differential evolution (DE)

1 Introduction

A study in computation science to find the procedures for the 'best' solutions is optimization. Optimization has been used widely in a various fields such as transportation, manufacturing, physics, and medicine [1, 2]. Problems associated with real-world optimization [3] are:

- Problems in differentiating between global optimal and local optimal solutions.
- Noise presence in evaluating the solutions.
- With the problem aspect, the 'jinx of dimensionality' roots the scope of the search space to propagate exponentially.
- Issues related with specified limitations and compulsion.

Various orthodox patterns of optimization have been suggested and established, and implementation/applications is also been observed, going from designing new drug or prediction of protein structure or to scheduling power system. These techniques are also facing problems in achieving the new emergent needs of current industry, where the prevailing optimization problems lean toward to be constrained, vigorous, multivariate, multi-objective and modalities [4, 5]. The existing and conventional optimization methods when attempt to highly nonlinear optimization task have been delimited by a frail global search capability, uncertainty, and inadequacy. Additionally with practical large-scale systems, various conventional optimization methodologies are not proficient to be implemented.

On the basis of qualities, we can divide optima into global or local optima. Figure 1 explains a belittlement issue $F \subseteq S$ in the viable search space. For minimization problems, we can define global and local optima as:

- Global Minima: objective function $f(x)$, solution $x^* \in F$ is global optima, if

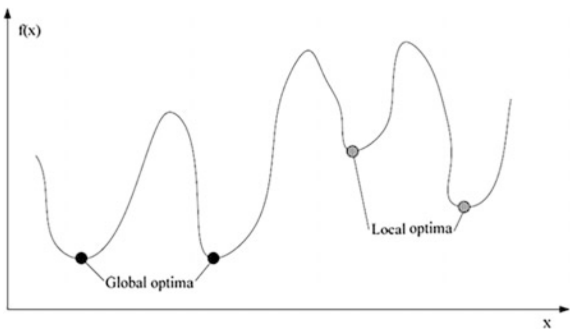

Fig. 1 Types of optima

$$f(x^*) < f(x) \because \forall x \in F \tag{1}$$

where $F \subseteq S$

- Local Optima: for the objective function $f(x)$ the solution $x_N^* \in \mathrm{N} \subseteq F$

$$f(x_N^*) < f(x) \forall x \in N \tag{2}$$

where $N \subseteq F$.

2 Evolutionary Algorithm

The development of evolutionary algorithms in last few years is very important amid all other search and optimization procedures. EAs [6] are a division of evolutionary computation, are a set of meta heuristics, population based used effectively in various applications having prodigious intricacy [7]. It success on solving difficult problems has been the engine of a field known as evolutionary computation (EC).

Evolutionary algorithms are based on the principles of adaptation and natural evolution of Charles Darwin [8] in 'On the Origin of Species'. Evolutionary algorithms (EAs) are pertinent to handle a varied kind of complex problems because of their ease, adequate tractability, and overall usability. There are few important characteristics which this evolutionary algorithm owns, and those can support them in belonging to the group of generated and test approaches:

- EAs are based on population, viz. these algorithms are a simultaneous collection of candidate solutions.
- These algorithms typically use amalgamation of one of more candidate solutions info into a current one.
- These algorithms are stochastic in nature.

A population to resolve for the optimization task is to be primed. Mutation and/or crossover are applied to generate new and different solutions. On the basis of resultant's fitness, appropriate selection approach is enforced to decide whether the selections of solutions are to be sustained into the subsequent generation. This process is then repeated to get the best fitness solution.

Most of the existent implementation of EA originates from any one of these 3 basic types: genetic algorithm (GA) [9], evolutionary programming (EP), and evolutionary strategies [10, 11]. In general, EAs is categorized by three facts:

- A set of solution candidate is sustained, which
- Goes through a selection procedure and
- Is operated by genetic operators typically recombination and mutation.

3 Hybrid Modal

It may not be adequate to find the preferred resolution of the problem from a simple and direct evolutionary algorithm for various problems. Previous studies suggest that a straight evolutionary algorithm might be unsuccessful to obtain a desired optimum solution for numerous types of problems which clearly demand the necessity for fusion of evolutionary algorithms with other probing optimization algorithms. Hybridization is usually done to mend:

- Performance of evolutionary algorithm
- Quality of the optimal results
- And as a part of higher structure, evolutionary algorithm can be integrated.

A number of studies in the past are being done on hybrid evolutionary optimization, and it has been observed the prodigious attainment of these techniques that can commendably conflict with their distinct shortcomings profiting from individual's strong points. The aims of developing fusion tactics are to handle very specific types of optimization problems. For instance, to resolve one of the most significant power system optimization glitches known as the unit commitment (UC) scheduling, a fusion of genetic algorithm (GA) and differential evolution (DE), termed hGADE has been proposed [12]. Fusion of the ACO, PSO, and 3-Opt algorithms can result in a hybrid evolutionary algorithm for solving traveling salesman problem [13]. Hybrid linkage crossover (HLX) is incorporated into differential algorithm (DE) to alleviate the disadvantages of DE and improve its performance [14]. To resolve multi-robot path planning, meta heuristic algorithms such as ACO-GA [15] and tree structure encoding-based hybrid EA [16] are used. Fusion of improved particle swarm optimization (IPSO) with an improved gravitational search algorithm (IGSA) has been proposed to determine the optimum route of the path for multi-robot in a muddle environment [17]. To precis; the hybridization inspiration is to improve reliability, conjunction hastening, and heftiness. Hybridization of evolutionary algorithm in general can be categorized into various categories as per the techniques or procedures, for example, hybridization motivation and hybridization design. To clarify, this can be divided into 'pre-processors and post-processors,' 'co-operators,' and 'implanted operators' built on the affiliation between all the nature-inspired computation methods involved. Essentially, a cautious and broad exploration of the taxonomy of hybridization would not only benefit us in gaining a profound considerate for the nature-inspired computation techniques but also elect the preeminent amalgamations for the optimization problems targeted.

Evolutionary techniques stake many resemblances, like adaptation, learning, and evolution. Alternatively, these techniques also have some distinctive dissimilarity, and individually have their individual benefits and drawbacks [18]. Advantages and disadvantages of evolutionary algorithms are summarized in Table 1. For example, differential algorithm (DE) requires high computational effort but the search results are effective.

Table 1 Advantages and disadvantages of the evolutionary algorithm

Evolutionary algorithm	Advantages	Disadvantages
DE	Active search	High computational work
PSO	Distribution of information	Hasty
MEC	Timely	Sluggish
SA	Heftiness	Long computation time
ACO	Pheromone-based exclusiveness	Over similarity
CSA	Multiplicity	Sluggish convergence speed
HS	Algorithm simplicity	Obsolete information

A. Motivation of Hybridization

Hybridization techniques are superior to standalone algorithm as these techniques have the competency of overpowering the shortcomings of standalone algorithms without losing their benefits. Hybridization of evolutionary algorithm can be done by several ways, and few of them are summarized below:

- The solutions can be created by problem-specific heuristics for the initial population.
- Obtained solutions from EAs can be enhanced by local improvement search procedure.
- Genotype solutions are mapped to phenotype solution by the algorithm and present the solution in indirect way.
- Problem knowledge is exploited by variation operator.

B. Hybridization Architecture

As represented in below diagram (Fig. 2), as per the nature of evolutionary algorithms we can divide hybrid nature-inspired algorithms into three groups.

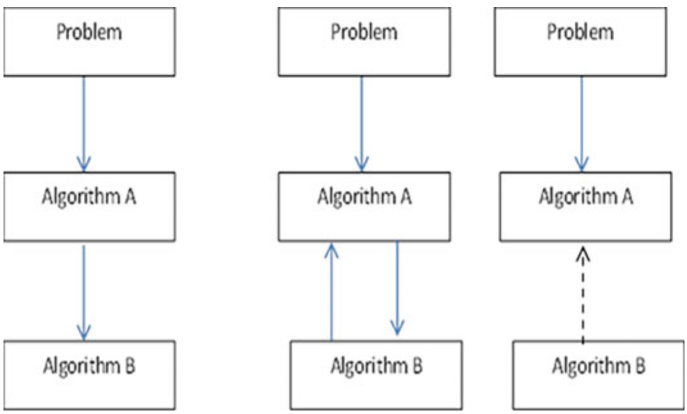

Fig. 2 Architectures of the hybrid NIC algorithms

- Pre-processors and post-processors: This type of hybridization NIC classification is the utmost prevalent and common, and in this type the optimization procedures are executed in sequence, i.e., results obtained by Algorithm 1 (pre-processor) are tweaked by Algorithm 2 (post-processor). For instance, fusion of algorithm based on particle swarm optimization (PSO) algorithm and ant colony optimization (ACO) algorithm called PSO-ACO is proposed to inherit their advantages and overcome their drawbacks [19].
- Co-operators (Fig. 2): This is the type of hybrid association where two algorithms are involved in simultaneous to adjust each other output as shown in Fig. 2. Both the algorithms share the common information during the process of search. As an example, fusion of fuzzy inference and the evolutionary computation technique is done for the approximation of nonlinear function. Parameter of both the algorithms can be enhanced, tweaked, and controlled by them [20, 21].
- Embedded operators (Fig. 2): These types of technique are categorized by their design and architectures, where one algorithm is entrenched inside another algorithm. Usually in this type of approach from different optimization techniques, the local search and global search are combined together so as to enhance the hybridization convergence. As an example, in [22–24], to detect premature detection of high quality solutions, the heuristic local search strategies are engaged in the algorithm and then to further refine the pheromone concentration or to generate the new solution, the given solution by local search strategy can be applied.

To enhance the generic efficiency of evolutionary algorithms, several different heuristics/meta heuristic procedures have been used. Following are the few of the common hybrid design architectures used:

- One evolutionary algorithm is hybridized with another.
- Evolutionary algorithm assisted by neural network.
- Evolutionary algorithm assisted by fuzzy logic.
- Evolutionary algorithm abetted by particle swarm optimization (PSO).
- Evolutionary algorithm abetted by ant colony optimization (ACO).
- Evolutionary algorithm abetted by bacterial foraging optimization.
- Fusion between evolutionary algorithm and other heuristics.

4 Conclusion

From the scientific literature/databases, it is evident that the hybridization of evolutionary algorithms is popular. In this paper, we discussed the numerous hybridization options for an evolutionary algorithm, and basic hybrid evolutionary design/architecture is presented that has evolved during the last years. Some of the popular and important hybrid architectures are also reviewed as reported in literature. In future the above discussed framework and architecture can be efficiently applied to develop hybrid NIC algorithm that will provide a various other ways of resolving the real-world problems more efficiently and quickly with accuracy.

References

1. Chan, K.Y., T.S. Dillon, and C.K. Kwong. 2011. Modeling of a Liquid Epoxy Molding Process Using a Particle Swarm Optimization-Base Fuzzy Regression Approach. *IEEE Transactions on Industrial Informatics* 7 (1): 148–158.
2. Tang, K.S., R.J. Yin, S. Kwong, and K.T. Ng. 2011. A Theoretical Development and Analysis of Jumping Gene Genetic Algorithm. *IEEE Transactions on Industrial Informatics* 7 (3): 408–418.
3. Spall, J.C. 2003. *Introduction to Stochastic Search and Optimization*. Hoboken, NJ: Wiley.
4. Deb, K. 2001. *Multiobjective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley.
5. Coello, C.A.C., D.A.V. Veldhuizen, and G.B. Lamont. 2002. *Evolutionary Algorithms for Solving Multiobjective Problems*. Norwell, MA: Kluwer.
6. Eiben, A.E., and J.E. Smith. 2003. *Introduction to Evolutionary Computation*, 175–188. Berlin: Springer.
7. Fonseca, C., and P. Fleming. 1995. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation* 3 (1): 1–16.
8. Jones, G., A. Robertson, C. Santimetvirul, and P. Willett. 1995. Non-hierarchic Document Clustering Using a Genetic Algorithm. *Information Research* 1 (1): 1.
9. Ahn, C. 2006. *Advances in Evolutionary Algorithms: Theory, Design and Practice*. New York: Springer.
10. Bäck, T., and H. Schwefel. 1993. An Overview of Evolutionary Algorithms for Parameter Optimization. *IEEE Transactions on Evolutionary Computation* 1 (1): 1–23.
11. Beyer, H.G., and B. Send Hoff. 2008. Covariance Matrix Adaptation Revisited—The CMSA Evolution Strategy. In *Lecture Notes in Computer Science*, 5199, 123–132.
12. Anupam, Trivedia, Dipti, Srinivasana, Subhodip, Biswasc, Thomas, Reindlb. 2015. Hybridizing Genetic Algorithm with Differential Evolution for Solving the Unit Commitment Scheduling Problem. *Science Direct Swarm and Evolutionary Computation* 23: 50–64.
13. Mahi, Mostafa, Ömer Kaan Baykan, and Halife Kodaz. 2015. A New Hybrid Method Based on Particle Swarm Optimization, Antcolony Optimization and 3-opt Algorithms for Traveling Salesman Problem. *Applied Soft Computing* 30: 484–490.
14. Cai, Yiqiao, and Jiahai Wang. 2015. Differential Evolution with Hybrid Linkage Crossover. *Information Sciences* 320: 244–287.
15. Oleiwi, B.K., H. Roth, and B.I. Kazem. 2014. A Hybrid Approach Based on ACO and GA for Multi Objective Mobile Robot Path Planning. *Applied Mechanics and Materials* 527: 203–212.
16. Ming-Yi Ju, Siao-En Wang, and Jian-Horn Guo. 2014. Path Planning Using a Hybrid Evolutionary Algorithm Based on Tree Structure Encoding. *The Scientific World Journal*.
17. Das, P.K., H.S. Behera, and B.K. Panigrahi. 2016. A Hybridization of an Improved Particle Swarm Optimization and Gravitational Search Algorithm for Multi-Robot Path Planning. *Swarm and Evolutionary Computation*.
18. Wolpert, D.H., and W.G. Macready. 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation* 1 (1): 67–82.

19. Jing, Yang, Wen-Tao, Li, Xiao-Wei, Shi, Li, Xin, Jian-Feng, Yu. 2013. A hybrid ABC-DE Algorithm and Its Application for Time-Modulated Arrays Pattern Synthesis. *IEEE Transactions on Antennas and Propagation* 61 (11).
20. Yu-Jun, Zheng, Hai-Feng, Ling, Sheng-Yong, Chen, Jin-Yun, Xue. 2015. A Hybrid Neuro-Fuzzy Network Based on Differential Biogeography-Based Optimization for Online Population Classification in Earthquakes. *IEEE Transactions on Fuzzy Systems* 23 (4).
21. Jie, Li, Junqi, Zhang, ChangJun, Jiang, and MengChu, Zhou. 2015. Composite Particle Swarm Optimizer with Historical Memory for Function Optimization. *IEEE Transactions on Cybernetics* 45 (10).
22. Lixin, Tang, and Xianpeng, Wang. 2013. A Hybrid Multi-Objective Evolutionary Algorithm for Multi-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation* 17 (1).
23. Liangjun, Ke, Qingfu, Zhang, and R. Battiti. 2014. Hybridization of Decomposition and Local Search for Multiobjective Optimization. *IEEE Transactions on Cybernetics* 44 (10).
24. Sindhya, K., K. Miettinen, and K. Deb. 2013. A Hybrid Framework for Evolutionary Multi-Objective Optimization. *IEEE Transactions on Evolutionary Computation* 17 (4): 495–511.