A Review of Bio-Inspired Computing Methods and Potential Applications

Amrita Chakraborty and Arpan Kumar Kar

Abstract The domain of bio-inspired computing is increasingly becoming important in today’s information era. More and more applications of these intelligent methods are being explored by information scientists for different contexts. While some studies are exploring the application of these algorithms, other studies are highlighting the improvement in the algorithms. In this study, we identify five more popular algorithms and briefly describe their scope. These methods are neural networks, genetic algorithm, ant colony optimization, particle swarm optimization, and artificial bee colony algorithms. We highlight under what context these algorithms are suitable and what objectives could be enabled by them. This would pave the path for studies conducted in the future to choose a suitable algorithm.

Keywords Bio-inspired computing · Neural networks · Genetic algorithm · Ant colony optimization · Particle swarm optimization · Artificial bee colony

1 Introduction

The domain of bio-inspired computing is gradually getting prominence in the current times. Algorithms from bio-inspired computing are gradually gaining prominence since these algorithms are more intelligent and can learn. Further, these algorithms have the capability to generate information based on the changes in the...
ecosystem that is generating data. The subsequent sections are subdivided in the following: first we identify the different types of popularly used algorithms among bio-inspired algorithms. Subsequently we explore the scope of these algorithms in specific context. Then based on the applications and scope of the algorithms, we try to provide insights on the potential applications for future research directions. We do not attempt to explore the scope or performance centric issues for the current study. Further the comparison through empirical validation is also beyond the scope of our current study, which has been built through the review of literature.

2 Review of Algorithms

This section is subdivided into independent reviews of multiple algorithms. All of these algorithms try to replicate the way biological organisms and entities operate to achieve high efficiency, even if sometimes the actual optimization is not achieved. After the review of algorithms, we focus on the scope of applications.

2.1 Neural Networks

Neural Networks [11] are often defined as adaptive nonlinear data processing algorithms that combine multiple processing units connected in a network in different layers. These networks are characterized by being self-adapting, self-organizing, and with the potential to learn based on inputs and feedbacks from the ecosystem within which it is operating. These neural networks try to replicate the way the neurons in any organism are coded to take inputs and through operations in a black box, to provide an output. The difference of outputs from desired results is sent back as feedback to improve the processing systems.

While there are different approaches with which neural networks are implemented, probably the simplest implementation is that of a perceptron network where there is a feedback to improve upon the output and there is often a single layer that provides the internal operations. Perceptron networks can be used both for linear and nonlinear systems [22]. Further such a network could also have multiple inputs and multiple layers [2]. Also developments in neural networks have seen applications of probabilistic and approximation-based algorithms to accommodate imprecise or incomplete information to improve outcome [13, 24]. Also, the way information as processed was segregated into linear and nonlinear neural networks in the same way as the individual information processing nodes operated within the network [11, 20]. Recent literature [23] highlights how deep, shallow, unsupervised, supervised, and reinforcement-based approaches are used to train the
networks based on data availability. However, neural networks can be combined with other algorithms and methods, based on the needs of the problem to provide improved predicting capabilities to the system [14, 23].

2.2 Genetic Algorithm

Genetic algorithm [12] was introduced to accommodate the way nature uses powerful computational techniques to obtain a highly suitable solution. It is an evolutionary search heuristic that mimics the process of natural selection. It is used to solve a variety of single and multi-objective problems that are combinatorial in nature and NP hard [1, 4]. For using this algorithm, a problem solution is defined in terms of the fitness function where the fitness of the potential solution is an indicator of its suitability. This fitness may be represented by integers, vectors, matrices, linked lists, or other data structure based on how the problem is tackled. A coding mechanism may transfer the data structure to the fitness function. Fitness could be either a maximization or a minimization function, based on objective. To address this situation, four basic operators were defined in genetic algorithm literature [3, 21]. For a new candidate solution to be produced, a pair of pre-optimized solutions are modified by using these operators. By using the operators, a “child” solution is obtained through crossover and reproduction where typically the new candidate solution shares many of the characteristics of its “parents”. However in the mutation operator, a specific fitness driver may be changed to enhance the fitness of the candidate solution abruptly. This is done to avoid local optimality and challenges associated with intermediate levels [19, 25].

2.3 Ant Colony Optimization

Ant colony optimization [6, 7] is a heuristic search algorithm, for solving combinatorial optimization problems. It is based on the indirect communication of simple agents (called ants here) foraging for information, mediated by artificial pheromone trails. The trails serve as a distributed numerical information for the agents to construct solutions based on probabilistic search experience. The results are obtained in a reasonable amount of search time.

In this algorithm, the solution is often attempted in a sequence of iterative steps [5]. Candidate solutions are developed from a sample of solutions using a parametric probability distribution. For doing so, a group of ants is selected and a pool of decision variable is defined in the problem. The ants select the design variables for creating the candidate solutions. As the ants explore the candidate solutions, a
local updation of the solution is done based on its optimality. These candidate solutions are used to modify the value of the trails based on the local updation in such a way to select the higher quality solutions in the subsequent sampling for candidate solutions by the same group of ants. The candidate solutions created in the initial phase therefore pave the path for optimality.

2.4 Particle Swarm Optimization

Particle swarm optimization is a heuristic algorithm inspired from the collective group behavior of animals such as fish schooling, insect swarming, or birds flocking, whereby the group attempts to meet the collective objective of the group based on the feedback from the other members. It is used for problems where the function to be optimized is discontinuous, non-differentiable with too many nonlinearly related parameters [8]. Assumption for this algorithm is that the collective intelligence of the group is more than the sum of individual intelligence of its members. This algorithm operates in a sequence of few iterative steps [8, 9]. Each particle or member in the swarm (say a bird or fish) tries to sense a potential solution at any point of time. It communicates a signal proportional to the suitability of the candidate solution to the other particles in the swarm. Each swarm particle or member can therefore sense the strength of the signal communicated by the other members, and thus the suitability of the candidate solution based on a fitness function [9]. When a particle or member tries to focus on a more suitable candidate solution from among the locally available candidate solutions, based on different learning mechanisms [26, 27], a new movement direction is identified along with an inertial influence to gradually guide the particles toward an optimal solution wherever possible. Application of this algorithm could be in multi-criteria decision problems, searching, constraint-based optimization problems, deterministic optimization problems, scheduling problems, thresholding, and maximization/minimization problems.

2.5 Artificial Bee Colony Algorithm

The artificial bee colony algorithm [10, 15] is a metaheuristic-based optimization algorithm which searches for an optimal numerical solution among a large number of alternatives while trying to solve NP hard problems [16]. This approach is based on the foraging behavior of the honey bee swarm. The behavior of honey bees based on the communication, task allocation, nest site selection, reproduction, mating, floral foraging, and pheromone laying and navigation behaviors of the swarm has been used in modification of this algorithm [15, 16, 26].
In this algorithm [15, 16], first the vectors of the population are initialized as a potential food source. Employed bees search for new food sources with a random stimulus initially. Subsequently, once a food source is identified (a candidate solution), the fitness of the same is identified and computed. In the next phase, if a new food source is subsequently discovered (a new candidate solution) by “employed bees” with a greater fitness, the new source is adopted else and the new one is rejected. Employed bees share the fitness information with the onlooker bees who choose their food source based on the probability of the food occurring. If bees are unable to improve the fitness of the food source, their solutions are rejected.

3 Discussion on the Scope of Applications

In this section, we highlight the potential scope of applications based on the evidences in published literature. However, due to lack of page constraints, we are brief in our discussion. Neural networks [11, 14, 23, 24] have been used extensively for the generation of association rules, classification problems, detection problems, pattern recognition, nonlinear regression, feature selection, missing data prediction, time series prediction, data normalization, principal component analysis, and probabilistic prediction. Genetic algorithms [1, 3, 21] have been used for searching among alternatives, maximization/minimization problems like the traveling salesman problem, sorting problems, multi-objective decision-making, multi-criteria decision-making, and constrained optimization problems. Ant colony optimization [17, 18] is extended to be able to tackle mixed variable optimization problems and continuous optimization problems for selection-, searching-, and optimization-based problems. Application of particle swarm optimization [5–7] could be in multi-criteria decision problems, searching, constraint-based optimization problems, deterministic optimization problems, scheduling problems, thresholding, and maximization/minimization problems. Artificial Honey Bee algorithm [9, 15, 16] has predominantly been used in literature as a single objective numerical value optimizer. Further it has been used for searching, routing problem, assignment problem, allocation problem, and maximization or minimization problems.

4 Conclusion

While these algorithms have witnessed a lot of attention from information scientists in recent years, the understanding within the domain is far from being mature. Except for a few these algorithms which are selected for discussion here, literature presents a lot of debate on the convergence and stability of these algorithms. The focus of this paper is not to highlight how these algorithms may be used for solving real-life problem through scientific notation, but to provide initial food for thought for understanding the scope and objective for these algorithms, and then choose one
based on the scope of the research problem and explore it in greater depth. Such an exploration would require a detailed breakdown and coding of the individual systems (say the fitness functions or computational units) for the algorithm chosen for solving the research problem.

References