A Novel Classification for Evolutionary Algorithms

Puneet Joshi, Medha Joshi, Mohit Goswami
Assistant Professor, Dept. of Electrical Engg. Rajkiya Engg. College, AmbedkarNagar – 224122, Uttar Pradesh (India)
Assistant Professor, Dept. of Electrical Engg. SLSET Group of Institutions, Kichha – 263148, Uttarakhand (India)
Assistant Professor, Dept. of Mechanical Engg. Doon Institute of Engg. And Technology, Rishikesh -249204 Uttarakhand (India)

Abstract
Optimization is one of the most primitive tools of mathematics. Over the years, numerous algorithms have been developed that are classified based on their various attributes like number of objective functions, type of input data, etc. however, due to lack of critical categorization of these methods it becomes very tedious task to select any algorithm for a particular problem. Therefore, this paper proposes a novel classification for the meta-heuristics based methods that will serve as a valuable reference for both newbies and experts.

Keywords: Optimization, Evolutionary Algorithms, Meta-heuristics.

1 Introduction
One of the most substrative principles in our world is the quest for an optimal state. The search for optimal state can be observed in non-living things as well, whether they are atoms trying to minimize the energy of their electrons or solid bodies trying to assume energy-optimal crystal structures during freezing. The same goes for the biological principle of survival of the fittest, which, together with the biological evolution, leads to better adaptation of the species to their environment. Here, a local optimum is a well-adapted species that dominates all other animals in its surroundings.

As long as humankind exists, we strive for perfection in many areas. We want to reach a maximum degree of happiness with the least amount of effort. In our economy, profit and sales must be maximized and costs should be as low as possible. Therefore, optimization is one of the oldest of sciences, which even extends into daily life.

If something is important, general, and abstract enough, there is always a mathematical discipline dealing with it. Global optimization is the branch of applied mathematics and numerical analysis that focuses on, well, optimization. The standard form of a continuous optimization problem is:

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g_i(x) \leq 0 \quad i = 1,\ldots,m \\
\end{align*}
\]

where,

\[
f(x) \text{ is the objective function that is to be minimized over the } n \text{-variable vector } x, \quad g(x) \text{ are called the equality constraints, } h(x) \text{ are called the inequality constraints, } m \geq 0, p \geq 0, \text{ if } m \text{ and } p \text{ are equal to } 0 \text{ then the problem is termed as the unconstrained optimization problem.}
\]

By convention, the standard form defines a minimization problem. A maximization problem can be treated by negating the objective function. Over the years, numerous optimization algorithms have been proposed which find their use in diverse fields and problems. Despite the significant number of recently proposed algorithms in this field, a fundamental question, worth mentioning is that whether there exists an approach capable of solving all optimization problems. However, according to the No-Free-Lunch (NFL) theorem there is no single algorithm for solving all optimization problems. This means that an algorithm may perform well in some situations but fail to solve different set of problems. Therefore, everyday improvements in the current algorithms or new and more efficient algorithms are been proposed.

2 Classification of Optimization Techniques
Thus, out of the numerous methods available in literature selecting a particular method for the given problem becomes a cumbersome process. This problem is somewhat simplified with a well-defined classification that can present a critical comparison amongst the methods. Some of the important classifications based on different factors/characteristics are:

- **Based on the availability of data**
  - Deterministic algorithms and Probabilistic optimization algorithms
- **Based on the values of variables**
  - Continuous Optimization and Discrete optimization
- **Based on the constraints**
  - Constrained and Unconstrained Optimization
- **Based on number of Objective functions**
  - None, One or Many Objectives
- **Based on the optimization speed**
  - Online optimization and Offline optimization.

Deterministic Optimization is based on the
assumption that the data for a given problem is known. However, in real life problems the accurate data cannot be known with certainty. There may be some measurement error or the data may represent some futuristic information such as price or product demand that cannot be definite in nature.

**Probabilistic Optimization** is based on the fact that the actual data for any given problem is uncertain and the probabilistic distributions are used to govern or estimate the data. In this optimization, a feasible probabilistic distribution policy for almost all data instances is established and then used for the optimization of the performance of the problem.

**Continuous Optimization** is a classification, which is based on the values of a variable. For continuous optimization the variables can have any real values. These variables are known as continuous variables.

**Discrete Optimization** uses the values of variables from a discrete set such as a subset of integers. These variables are known as discrete variables. Discrete optimization problems are considered harder than continuous optimization problems and with the advancement of computing technology and improved algorithms, the size and complexity of discrete optimization problems have also increased.

**Constrained Optimization** deals with the problems that have explicit constraints on the variables. These constraints on the variables can be in the form of simple bounds as well as the system of equalities and inequalities. The nature of constraints and the smoothness of functions is also specified to further classify the constrained optimization problem.

**Unconstrained Optimization** problems do not have any constraint on their variables. This optimization finds its use in many practical applications and it can also be used for the reformulation of constrained optimization problems where a penalty term is used to replace the constraints in the objective function.

There are many other optimization technique classifications such as: Divide and Conquer algorithms, Branch and Bound algorithms, Gradient Search, Greedy algorithms etc. but all of these classifications use a single characteristic to define the optimization processes such as classification based on objective function, based on constraints etc.

3 History of Optimization

There is not a fixed point of time that denotes the inception of optimization techniques but it can be estimated to have evolved along with the advent of calculus (either by Fermat or Descartes). There are many optimization problems which were investigated in 17th and 18th centuries along with the development of calculus of variations such as Transportation problem, Secretary problem, Plateau's problem, Brachistochrone’s problem etc. and various researchers such as Newton, Leibniz, Bernoulli, Euler and many others had contributed in solving these numerous real life optimization problems. Newton Raphson method can be said as one of the first iterative optimization techniques but it also took a long time for this technique to become what we see it today. After that came the method of Least Square Approximation which was developed by Gauss and he used it to predict the location of Ceres (a newly found asteroid then). One other important method that was developed in 19th century was the method of Gradient Descent by Cauchy (1847). However, it provided slow convergence as compared with Newton Raphson.

In 20th century, several optimization techniques were developed with further development of Calculus of Variations. The very first book on Optimization, Theory of Maxima and Minima by Harris Hancock, was published in 1917. A general formulation of Travelling Salesman Problem (TSP) was given by K. Menger in 1932. Russian mathematician L. V. Kantorovich developed an important technique of optimization known as Linear Programming (LP) or Linear Optimization in 1939. This method was useful for the optimization of linear objective functions with linear constraints. The LP was used by Neumann to prove a Min–Max method or Theory of Duality which was also used in the Game Theory by J. Von Neumann. J. Von Neumann is also considered an important person behind the development of Operations Research. The Simplex method (1947) was also developed around the same time as LP. Non linear programming was also developed in the same year as LP by W. Karush (1939) but it was only accepted later when it was seen in a paper presented by Kuhn–Tucker in 1951. In the upcoming years a method known as Dynamic Programming was given by Richard Bellman (1952) to solve the complex problems by breaking them into simpler problems. The methods that we have discussed until now were developed to be applied over known differentiable functions and constraints and these methods fall under the category of deterministic optimization. In 1951, American mathematician H. E. Robbins and S. Monro introduced Stochastic Optimization which was useful for the purpose of finding the maxima or minima of some random and unknown function with unknown derivatives and random constraints. However, these Stochastic Approximation methods could not go beyond a local optimum, when they were applied over the multi model functions, same as other deterministic algorithms. To overcome this limitation, Probabilistic Meta-heuristics were developed. Simulated Annealing was one of these methods and it was also first of its kind which could converge to the global optimum in large search spaces of any function. With the advent of computers solving optimization problems became easier and the concept of Artificial Intelligence came into light, and thus, various new optimization algorithms were developed. A method named as Direct Search method was developed which worked on the concept of pattern search by exploration in search space and it was better than random search methods.

Another method named as Nelder-Mead method or Modified Simplex Method became quite popular during 1970s and is still considered as one of the most popular heuristic optimization techniques in use. The concept of Artificial Intelligence gave rise to the development of Evolutionary Algorithms which were based on stochastic meta-heuristic optimization technique. These Evolutionary Algorithms were inspired by the biological evolution of species such as reproduction, mutation, recombination and selection. In 1962, John Holland introduced a new algorithm in
the series of Evolutionary Algorithms known as Genetic Algorithm by proposing crossover and other recombination operators. As the years passed, GA found its use in a broad range of fields i.e. engineering problems, structural optimization and many abstract mathematical problems. Then came Differential Evolution (DE) by Rainer Storn and Kenneth Price in 1995, which was the most recent in the category of EA. Around the same time, one of the most successful probabilistic meta-heuristic method, Particle Swarm optimization (PSO), was developed and proposed by Russell C. Eberhart and James Kennedy. Therefore, many optimization techniques have been developed until date and remains a promising research domain.

4 Meta-Heuristics Methods
A metaheuristic is a high-level problem independent rubric to develop heuristic optimization algorithms. The term was coined by Glover in 1986 and combines the Greek prefix meta meaning high-level with heuristic meaning to search. Heuristic means local search, metaheuristic means generalized local search. Notable examples of metaheuristics include Genetic/ Evolutionary algorithms, Simulated Annealing, Tabu search, adaptive neighbourhood search, variable neighbourhood search, and Ant colony optimization, etc.

Meta-heuristics algorithms attempt to find the best (feasible) solution out of all possible solutions of an optimization problem by performing a series of operations. Based on the way meta-heuristics change the solutions, they can be distinguished in three fundamental classes.

Local search meta-heuristics iteratively make small changes to a single solution (i.e. current solution) and improve it with each iteration. In each iteration, the current solution is replaced by a solution from its neighbourhood using the move or search strategy.

Constructive meta-heuristics construct solutions from their constituting parts by adding one element at a time to a partial solution.

Population-based meta-heuristics iteratively select and combine solutions from a set, called population, into new one. However, these classes are not mutually exclusive and many meta-heuristic algorithms combine ideas from different classes and these are called Hybrid meta-heuristics.

5 Proposed Classification
In present times, most of the meta-heuristic methods are being developed by using various natural or man-made processes (bacterial foraging, river formation, biogeography, musicians playing together, electromagnetism, gravity, colonization by an empire, mine blasts, league championships, clouds etc.) as their framework. There are also a lot of novel meta-heuristic methods which are inspired by the behaviour of animals (Ants, bees, bats, wolves, cats, fireflies, eagles, vultures, dolphins, frogs, salmon, vultures, termites, flies etc.).

Thus, out of the numerous methods available in literature selecting a method best suited for the problem in hand becomes a challenging task. This problem is simplified with the, henceforth, proposed classification criteria that can present a critical comparison amongst the methods.

No. of tuning parameters
Every meta-heuristics based approach has some parameters that govern the exploration/exploitation capability of the method. For example: In a PSO, c1, c2 and w are the parameters that need to be tuned for a particular problem. Determining these parameters is an exhaustive as it is a hit and trial process. Therefore, it would be reasonable and effective to select a method that has lesser number of tuning parameters.

Memory Requirement
Some methods require the solution of the previous (n-1)th iteration to improve the next nth iteration termed as vectored methods. Whereas, some non-vectored methods improve the solution of the nth iteration directly. For example: PSO requires (n-1)th solution to generate nth solution whereas GSA does not need (n-1)th solution to generate nth solution. In this way, PSO requires more memory than GSA.

Exploration-Exploitation Capability
Exploration is the measure of the capability of a method to search for the optimum solution in the entire search space. Exploitation is the measure of the capability of a method to search for the optimum solution in the neighbourhood of a particular solution. Notably, Exploration and Exploitation are the two cornerstones and improving one results in degradation of the other. Thus, first the requirement for the problem should be analyzed that whether a better exploration or exploitation is required. Accordingly, a method may be selected. For example: As indicated in literature, PSO has a better exploration capability than GSA. Thus, if a problem demands a rigorous exploration then PSO may be preferred.

Conceptual/Programming Complexity
Some methods like PSO, ACO, GA are easy to understand and program, compared to methods like Ant Lion, Coral Reef, etc. Thus, to develop the conceptual understanding the beginners may avoid such methods.

Population Size
Some methods require large population size for searching purpose and, therefore, may not be suited to practical applications where a small population size is necessary.

Execution Time
The time taken by a method takes to find the optimum solution depends on two factors: (a) Population size (b) Program length
Consider, a practical example, where you have to find the optimum duty cycle for a DC-DC Converter so that maximum power is extracted from the source. Thus, some random initial population (duty cycle) has to be generated and power corresponding to each duty cycle is to be sensed from the system and only then rules to generate new population can be applied. Thus, we have to wait, until the system stabilizes, for each duty cycle in order to sense its corresponding power. Thus,
larger the population size more is the execution time. Though, nowadays, the processing speed for any hardware is in the range of milliseconds to nanoseconds, therefore, program length may not be a very influential parameter; nonetheless, the execution speed does depend on it.

**Nature of the Input Data**
The data for any problem may be binary, real no. or complex number in nature. Hence, a suitable choice amongst the methods is to be made accordingly.

**Hardware Implementation**
Some methods require cheaper hardware like microcontrollers; however, some methods require high fidelity hardware like FPGA, DSPs etc. for hardware realization.

**6 Conclusion**
Thus, a novel classification methodology for optimization algorithms has been proposed that would be beneficial and efficient for beginners as well as experienced users to narrow down to a particular optimization technique best suited for the problem.

**References**
14. S. Kiranyaz, T. Ince, and M. Gabbouj, Multidimensional Particle Swarm Optimization