№ 4 (2), 2020 ISSN: 2682 – 9517 (print) ISSN: 2683 – 0930 (online)
INSTITUTE OF INFORMATICS AND INNOVATIVE TECHNOLOGIES

Stochastic Optimization Algorithms.Why?

Tarun Kumar Sharma[1] and Jitendra Rajpurohit[2]

[1]Department of Computer Science & Engineering and [2]Department of Computer Science
[1]Shobhit University Gangoh, Saharanpur and [2]University of Petroleum and Energy Studies, INDIA
[1]tarun.sharma@shobhituniversity.ac.in; [2]jiten_rajpurohit@yahoo.com

***Abstract:* — Most of the real world optimization problems are highly complicated or intrinsic complex and may not be solved using well know deterministic methods. The optimization problems are characterized as discrete or continuous; single objective or multi-objective and constrained or unconstrained. The aim is to find out most favorable solution/output and simultaneously reducing the input cost. This is how Swarm Intelligent algorithms (SIA) came into picture. SIA not only helps in obtaining near optimal solution but also at low computational cost. In the present study a brief description on the SIA is presented to help the novel readers.**

Keywords:* — *Swarm Intelligent Algorithms, Nature inspired; Bio Inspired; Optimization; Artificial Intelligence

I. INTRODUCTION

Some computational problems may have multiple solutions. An acceptable solution is one which provides the output better in terms of cost incurred in providing input. The aim of optimization is to find out most favorable output and simultaneously reducing the input cost. Putting in simple words, Optimization is the process of finding the best suitable solution from the available set of solutions to a given problem. Optimization problems are not that simple always but a lot of complex issues are associated with it. A little complex optimization problem Optimal Capacity of Gas Production Facility is detailed below [1] (Rao, 1996):

$$f(m) = 61.8 + 5.72m_1 + 0.2623 \times$$
$$\times \left[(40 - m_1) \times ln\left(\frac{m_2}{200}\right)\right]^{-0.85} + 0.087 \times$$
$$\times (40 - m_1) \times ln\left(\frac{m_2}{200}\right) + 700.23m_2^{-0.75}. \quad (1)$$

Subject to: $m_1 \geq 17.5$; $m_2 \geq 200$;

$17.5 \leq m_1 \leq 40$; $300 \leq m_2 \leq 600$

Some basic terms related to optimization are being discussed below with reference to

- **Input variables:** These are the variables that affect the value of the out coming solution. Changes incurred in the outcome for each of the variable may vary but for the purpose of optimization, each variable necessarily affects the output. m_1 and m_2 are the input variables
- **Search space:** Search space is the allowed range of input variables. Individual variables may have separate ranges. The range of variables m_1 and m_2 in Example 1.1 is [17.5, 40] and [300, 600] respectively. In a multi-dimensional problem, every point in the search space is composed of multiple points, each point has a value for each of the input variables in the specified range. Purpose of an optimization algorithm is to find out the point in the search space which provides the optimum value for the objective function.
- **Objective Function:** This is the function that needs to be optimized (Eq. (1)). Optimization problems are modelled either as minimization or maximization problems [2]. Problem discussed above is a minimization problem where f(m) is the cost function and needs to be minimized. Value of the objective function is also used as a metric to evaluate the performance of an optimization algorithm. More favourable value an algorithm provides, more accurate it is.
- **Constraints:** Constraints are the restrictions on the search space [2]. Simple constraints may be in the form of a one sided restriction on the value of an input variable while complex constraints may have a limitation on a combination of variable. Complex optimization problems may have multiple constraints.
- **Local Optima:** local maxima are points in the search space that provide local peak values of

the objective function. But these values are not desired as the purpose of the search during optimization algorithm is to find out either highest or the lowest peak value (global optima). Local optima create serious issues while searching for the global optimum. Many search algorithms are greedy in nature and conserves the best solution achieved at any stage of the execution. Once a local optimum is achieved, the search process becomes biased towards searching near the local optimum and at the end of the execution it returns one of the local optima as the best solution. Fig. 1 is a graph of such a function with multiple optima.

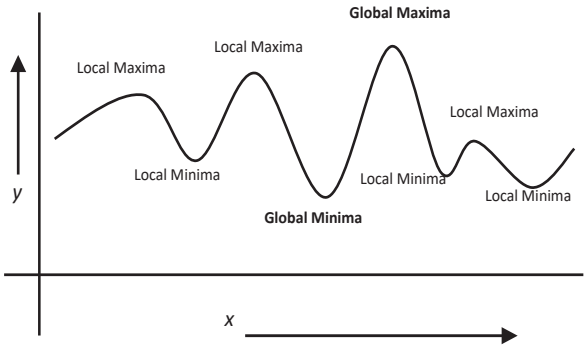

Fig. 1. Pictorial representation of local and global optima

- **Dimensions:** Number of input variables in an optimization problem is known as the dimensions or decision variables of the problem. Although, the complexity of an algorithm is not directly related to the dimensions of a problem, still very high dimensions increases the computational cost at least in terms of pre-processing and overhead. Number of dimensions in the problem taken is 2 as it has two input variables m_1 and m_2.

II. Deterministic vs Stochastic Optimization

Deterministic optimization problems are those where the output can be clearly determined from the given set of input variables. This relationship does not have any randomness contained in it. Every time the same set of input will provide the same set of outputs. Also, there exists a convergence proof for a deterministic problem. Few examples of deterministic optimization problems are as follows [3]:

— Linear Programming Problems (LPP)
— Mixed Integer Linear Programming Problems (MILP)
— Non-linear Programming Problems (NLPP)
— Mixed-integer non-linear programming problems (MINLP).

As the output is unique with respect to the set of inputs, a single run of deterministic optimization algorithms is execution is sufficient to find the optimized solution.

Stochastic Optimization is used for problems which do not have deterministic relationship between the set of inputs and outputs. Stochastic Optimization Algorithms use random variables that add randomness to the solutions. Each of the time a Stochastic Algorithm is executed with the same set of inputs, the randomness factor creates a different output. To counter the effect of randomness, the stochastic algorithms are repeated for multiple runs and then some statistical tool is used to predict the output. As there is no known deterministic relationship between input and output, a stochastic optimization problem is essentially a search process that chooses a point in the search space and evaluates the value of objective function at that point. This search is associated with randomness and some search mechanism. For many problems, specifically continuous optimization problems, it is not feasible to evaluate each point in the search space. It is possible that the solution provided by a stochastic algorithm is not the optimized one. So, stochastic optimization algorithms do not guarantee to provide the optimum solution but provides near optimal solutions. Practically, stochastic algorithms are accurate enough and provide acceptable solution. Deterministic problems can also be modelled as stochastic problems.

Algorithms used for optimization of deterministic and stochastic optimization are known as deterministic optimization algorithms and stochastic optimization algorithms respectively. The differences between deterministic and stochastic optimization algorithms are provided in Table 1.

III. Stochastic Optimization Algorithms

From the above discussions, it can be noted that a stochastic algorithm is applicable to almost all the optimization problems regardless of the nature of the problem. It is the only model to solve the problems where the relationship between input and output is not deterministic and the deterministic algorithms fail. Few major terms and properties related to stochastic optimization algorithms are discussed next:

Table 1. Major differences between Deterministic and Stochastic Algorithms [4]

Deterministic Algorithms	Stochastic Algorithms
Simple Linear problems	Complex Non-linear problems
Outputs are unique to a set of inputs	Same set of inputs may lead to different outputs in different runs
Knowledge of problem definition is required	Knowledge of problem definition is not required
Deterministic in nature. No randomness	Uses random variables
Only one execution is sufficient to find the optimum value	Are executed for multiple runs and then statistical tools are used to approximate optimum solution
Guarantee to find out the optimum	Do not guarantee that the solution found is the optimum but provides near optimum solution
Not applicable for a stochastic problem	Is applicable for deterministic problems as well as a special case.
No uncertainty in the model	Always a component of uncertainty and randomness
Convergence proof exists	Convergence proof does not exist

A. Heuristic

Stochastic algorithms are all search methods. They generate an individual random solution, test its fitness based on the objective function (also called fitness function) and based upon the fitness value decides the further course of action. Direction of this search depends upon the fitness value of the individual solutions being considered by the algorithm currently. Heuristic is the guiding tool that guides this search process. So, heuristic guides the search process of a stochastic method to generate the new points in the feasible space depending upon the current status of the search. Normally, the heuristic keeps the direction of the search towards best solution found so far with some added randomness.

B. Metaheuristic

Metaheuristic [5] is a class of high level problem independent set of optimization methods that do not need any specific details about the search surface. Metaheuristic can be seen as a combination of heuristics and some other algorithm specific optimization tools. They use black box methods for optimization and do not need much information about the optimization problem and are applicable to all the optimization problems but more efficient on non-deterministic problems. Most of the stochastic optimization algorithms are metaheuristic in nature.

C. Nature Inspired Optimization

Nature is the best source of optimization methods. There are species that are fighting for their survival for millions of years to the continuously changing environment, yet they are able to survive. Many natural species use methods of optimization while they search for food, trap a prey, confront a hunter species, searching for shelter, impressing their mate and alike. These species have evolved for millions of years and according to Darwin's theory [6], they are the best following the "survival of the fittest" principle. Their routine life involves many processes that have been evolving for millions of years and are optimized. There are many stochastic optimization methods that are majorly inspired by species such as ant, honey-bee, frog and alike. Examples of algorithms inspired by species are Ant Colony Optimization, Artificial Bee Colony, Shuffled Frog Leaping Algorithms, Whale Optimization, Spider Monkey Optimization etc.

Other than the living species, stochastic optimization also finds inspiration from some natural phenomena. The examples of such algorithms include Big bang-big crunch, Biogeography Based Optimization, Coral Reefs Optimization, Galaxy-based Search Algorithm, Lightning Search Algorithm, Swallow Swarm Optimization Algorithm, Virus Optimization Algorithm, Water Cycle Algorithm, Wind Driven Optimization and alike.

In addition to this, there is long list of stochastic optimization algorithms that are inspired by man-made artificial processes like Brain Storm Optimization, Greedy Politics Optimization, Grenade Explosion Method, Group Counselling Optimization, Group Search Optimizer, Interior Design and Decoration, Soccer Game Optimization, Society and Civilization, Teaching-learning based Optimization, FIFA World Cup Optimization etc.

Not only this, many stochastic search process are inspired by physical and chemical phenomena as well such optimization methods include Central Force Optimization, Charged System Search, Crystal Energy Optimization, Colliding Bodies Optimization, Electro-magnetism Optimization, Gases Brownian Motion Optimization.

A detailed survey of above mentioned algorithms can be found in [7].

Robustness and accuracy of nature inspired algorithms are inspiring the development of more and more such algorithms. A close observation to a natural phenomenon and then a lot of testing and algorithm specific inclusions can lead to a more efficient algorithm. Development process of a nature inspired optimization algorithm is illustrated in Fig. 2.

D. Categorization of optimization methods

Vast and varying needs of optimization have lead to development of a large number of optimization methods. Many of them are general and applicable to majority of optimization problems while most of them are specific to a certain class of optimization problems. It is really difficult to draw a clear and concise categorization of optimization approaches. Many categories of these approaches overlap on the basis of their working principle. So, many methods are fit for multiple categories. Yet, an attempt has been made to provide a categorization of optimization approaches in Fig. 2.

There is still scope for argument over this categorization.

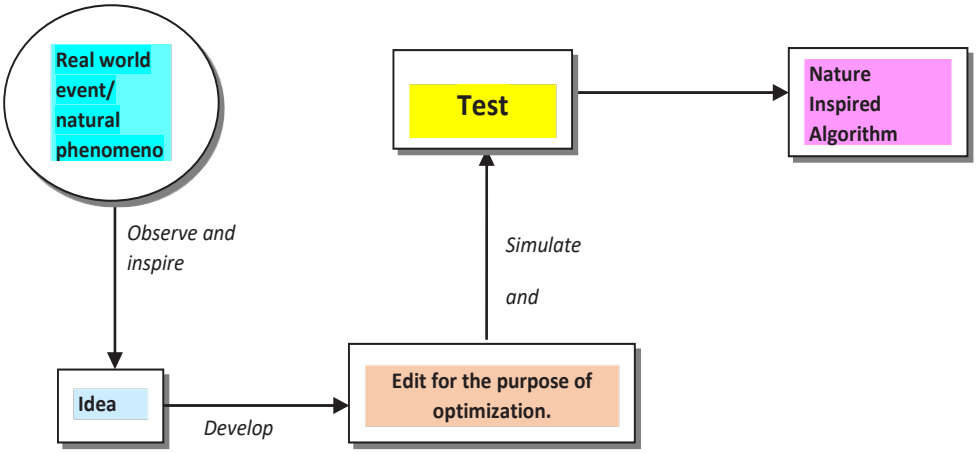

Fig. 2. Development process of a nature inspired optimization algorithm

E. Nature Inspired Metaheuristic Stochastic Optimization (NIMSO) Algorithms

As discussed earlier NIMSO algorithms are stochastic search methods and do not guarantee to provide the optimum but yet they are much popular. Reason for his popularity and acceptance is that they are the only class of methods for non-deterministic optimization problems. Most of these algorithms have very similar working principle. As the heuristic used in the method guides the direction of the search, even a small change in the working principle can change the direction of the search and the results may vary. The general framework of a NIMSO algorithm is presented in Fig. 2

Major steps involved in the working of a NIMSO algorithm including those shown in Fig. 3 are discussed below:

1) Initialization

Search agents are initialized randomly within the feasible search space. Group of these search agents is known as initial population of solutions. Number of the members in the population is known as population size (*n*). Population size generally remains same throughout the execution of the algorithm. Each dimension of a multi-dimensional problem may have different range. Population size is an important parameter of any NIMSO run but larger population does not necessarily enhance the performance of the algorithm.

2) Mutation, Crossover and other mixing operators

These are the operators that generate new sets of vectors from the mating pool. For the first iteration, the mating pool is formed from the initial population, after that it is formed from the off-springs obtained from the selection process. Mutation is the process that edits the value of one or more dimensions of vector in the form of search vector. Mutation operation used in basic Differential Algorithm (DE) [8] is shown in Eq. (2).

$$v_{i,g+1} = X_{r1,g} + F(X_{r2,g} - X_{r3,g}) \quad (2)$$

Where g is the generation (iteration) count.

r_1, r_2, r_3 are three randomly selected members of population and $r_1 \neq r_2 \neq r_3$.

$X_{new,g}$ is the new trial vector generated.

i is the population count and $1 \leq i \leq$ Population size.

Cross over replaces one or more dimension values among the members of the population. Many algorithms use hybrid methods of mutation and crossover and even many unique mixing operators to generate new off-springs. Crossover used in DE (Stone & Price, 1997) is shown in Eq. 3:

$$u_{ij,g+1} = \begin{cases} v_{ij,g+1}, if\ rand\ (j) \leq CR \\ X_{ij,g},\ rand\ (j) > CR \end{cases} \quad (3)$$

Where j is the dimension count and $j = 1$ to D.

CR is a pre-defined constant known as Crossover rate in the interval (0,1)

$Rand\ (j)$ is a random number in the interval (0,1)

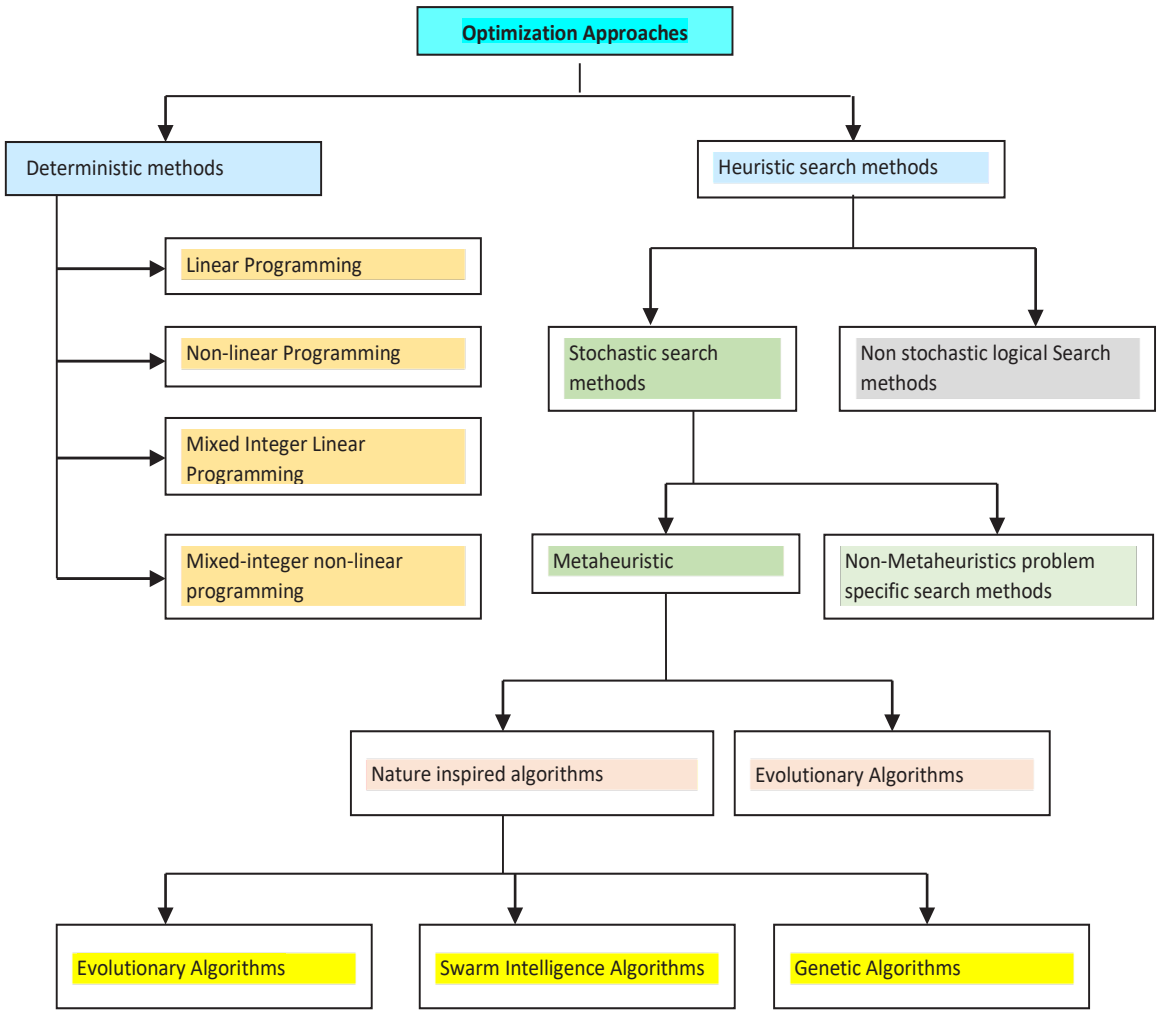

Fig 3. Categorization of optimization approaches

3) Off-springs

This is the set of trial points that are generated after the operations like mutation and crossover. These off-springs are to be tested for the individual values of objective function. Intermediate steps may have temporary high number of members of population but at the end of each iteration, generally, the population size remains constant. The set *u* created after the crossover operation shown in Eq. (3) is the set of off-springs. All the off-springs are then tested for their values of objective functions and few are selected to form the population for the next generation.

4) Selection

The process of selecting the members of population for the next generation is known as Selection. The principle followed is simply the survival of fittest. The

fittest members equal to the population size are selected to form the population for the next generation. Individual algorithms have slight variations in this selection process but the basic principle remains same. For example DE adopts a selection strategy where in a trial vector is selected for the next generation if and only if its fitness value is better than its corresponding target vector. The selection strategy used in basic DE is shown in Eq. 4:

Where f is the objective function for a minimization problem

Many NIMSO algorithms also use the fittest overall n members instead of comparison of individual trial and test vectors

$$X_{i,g+1} = \begin{cases} u_{i,g+1}, \ if \ f(u_{i,g+1}) \le f(X_i) \\ X_{i,g}, \ Otherwise \end{cases} \quad (4)$$

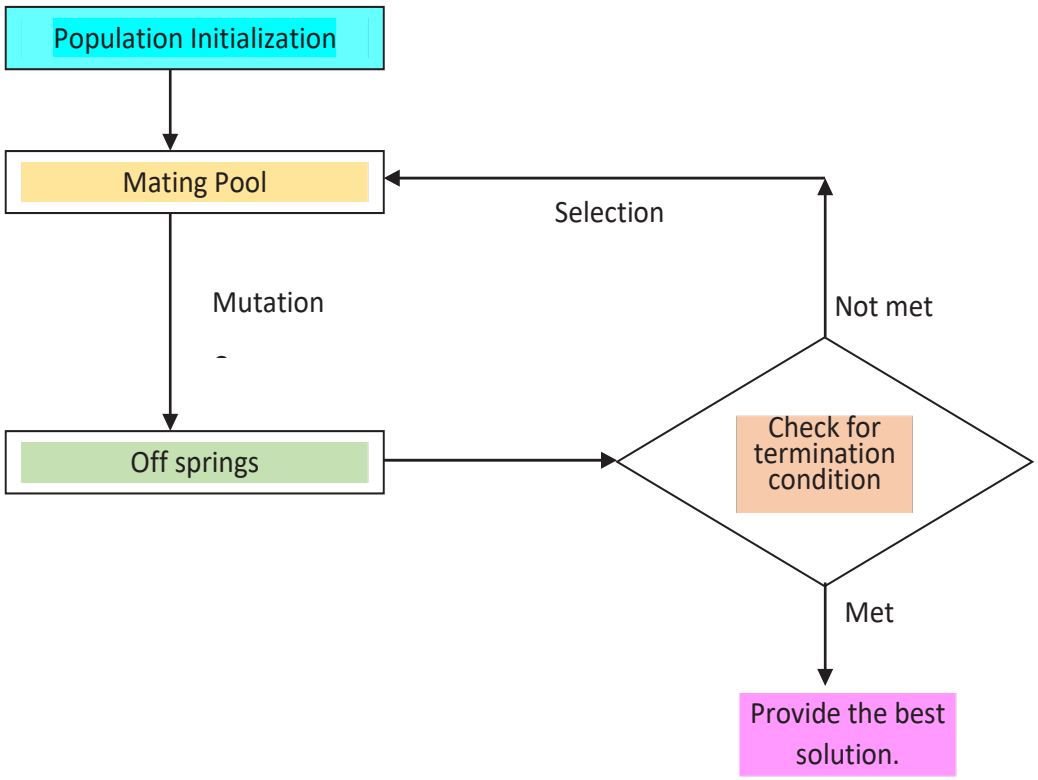

Fig 4. General framework of a NIMSO algorithm

5) Termination Condition

No NIMSO algorithm guarantees optimal solution. It is possible that a run of the algorithm can continue searching better solutions in every generation, in this way the number of generations can exceed to an unknown number. The execution has to stop at some point. There is always a termination condition set to stop the execution of a run of the algorithm. After every generation, this termination condition is checked, if met, the best solution obtained so far is considered as the optimum provided by the run. There may be many ways to set this termination condition. Following two termination criteria are used most often:

a) Acceptability of the solution:

NIMSO execution preserves the global best (gbest) value obtained so far at all the times. This global best keeps improving either keeps improving or at times become stagnant. If the termination condition is set according to the acceptability of the solution, the value of gbest is tested at the end of every iteration and whenever it reaches an acceptable solution, the execution is forced to stop.

For example, the sphere function mentioned in Eq. (5) is having its minima at 0.

The termination condition may be set as $gbest \le 10^{-4}$.

$$Minimize\ F(x) = \sum_{i=1}^{D} x_i^2 , \quad (5)$$

where $x_i \in [-100, 100]$ with optimum at $f(x)=0$ at $x_i = (0,0,0..)$.

b) *Difference vector approach*

Any new algorithm or variant is tested on a set of optimization functions with known optimum values to test them for their accuracy and efficiency. For such problems, the termination condition can be set as a threshold minimum value of the difference between the g_{best} and the known optimum value. *Vtr* is the most commonly used term for this threshold. For the same sphere problem shown in Eq. (1.6), the termination condition can $Vtr \leq 10^{-6}$, where $Vtr = g_{best} - 0$.

c) *Maximum value of iteration count*

An iteration is a significant point during the execution of an algorithm. Values of many parameters are tested after the end of iteration. These parameters include g_{best}, *Vtr*, number of iterations etc. Significant computational set-up has to be done at the start of iteration. A threshold maximum value of iteration count if quite often set as the termination condition.

d) *Maximum number of function evaluations (NFEs)*

Every trial vector has to be evaluated for its value of the objective function during the whole course of the NIMSO run. NFEs is considered as a measure of the computational cost incurred in a run of the of the NIMSO algorithm. A maximum threshold value of NFEs can also be set as the termination condition. The NFEs keep increasing even within a run, however, this condition is the comparison of NFEs and the maximum threshold value is done at the end of the iteration

A brief overview of over 177 algorithms is presented in paper [7]. Applications of the above discussed stochastic algorithms can be seen in [8] – [14].

IV. Conclusions

In this study, a brief overview of optimization algorithms and process of optimization is discussed along with the method to solve complicated belongs to the natural world. A brief overview of the formation of stochastic method is described. Also the working of stochastic algorithms is presented with the help of a mathematical equations along with the termination criterion.

References

[1] Rao Singiresu S. (1996), Engineering Optimization, Theory and Practice 3rd. Edition, chapter 1 Introduction to Optimization, Wiley and Sons, 1–64.

[2] Weise, Global Optimization Algorithms – Theory and Application. Germany: it-weise.de (self-published), 2009.

[3] Deb K. (2005) Multi-Objective Optimization. In: Burke E.K., Kendall G. (eds) Search Methodologies. Springer, Boston, MA

[4] Mario Francisco, Silvana Revollar, Pastora Vega, Rosalba Lamanna. (2005), A Comparative Study Of Deterministic And Stochastic Optimization Methods For Integrated Design Of Processes. IFAC Proceedings Volumes, Volume 38, Issue 1, 2005, Pages 335-340 https://doi.org/10.3182/20050703-6-CZ-1902.00917

[5] Yang XS. (2011) Metaheuristic Optimization: Algorithm Analysis and Open Problems. In: Pardalos P.M., Rebennack S. (eds) Experimental Algorithms. SEA 2011. Lecture Notes in Computer Science, vol 6630. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20662-7_2

[6] Paul, D.B. The selection of the "Survival of the Fittest". J Hist Biol 21, 411–424 (1988). https://doi.org/10.1007/BF00144089

[7] Jitendra Rajpurohit, Tarun K. Sharma, Ajith Abraham, Vaishali (2017).Glossary of Metaheuristic Algorithms. International Journal of Computer Information Systems and Industrial Management Applications, Vol. 9, pp. 181-205

[8] Bilal,Millie Pant,Hira Zaheer,Laura Garcia-Hernandez,Ajith Abraham (2020) Differential Evolution: A review of more than two decades of research. Engineering Applications of Artificial IntelligenceApril 2020 https://doi.org/10.1016/j.engappai.2020.103479

[9] Kashif Hussain, Mohd Najib Mohd Salleh, ShiCheng, Yuhui Shic, Rashid Naseemd. (2020) Artificial bee colony algorithm: A component-wise analysis using diversity measurement. Journal of King Saud University - Computer and Information Sciences, Volume 32, Issue 7, September 2020, Pages 794-808

[10] lDervis Karaboga, Bahriye Akay, Nurhan Karaboga (2020) A survey on the studies employing machine learning (ML) for enhancing artificial bee colony (ABC) optimization algorithm. Cogent Engineering Volume 7(1), https://doi.org/10.1080/23311916.2020.1855741

[11] M. Janga Reddy, D. Nagesh Kumar (2021) Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review. H2Open Journal, Vol.3 (1): 135–188. https://doi.org/10.2166/h2oj.2020.128

[12] H Gupta, S Kumar, D Yadav, OP Verma, TK Sharma, CW Ahn, JH Lee (2021) Data Analytics and Mathematical Modeling for Simulating the Dynamics of COVID-19 Epidemic—A Case Study of India. Electronics, MDPI, 10(2), 127

[13] Tarun K. Sharma (2020) Enhanced Butterfly Optimization Algorithm for Reliability Optimization Problems Journal of Ambient Intelligence and Humanized Computing https://doi.org/10.1007/s12652-020-02481-2

[14] Tarun K. Sharma and Ajith Abraham (2019) Artificial Bee Colony with Enhanced Food Locations for Solving Mechanical Engineering Design problems, Ambient Intelligence and Humanized Computing, Springer.