International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

NATURE INSPIRED COMPUTATIONAL INTELLIGENCE: A SURVEY

Diriba Kajela[1], Mukhdeep Singh Manshahia[1*]

1. Department of Mathematics, Punjabi University Patiala, Punjab, India.
Corresponding author email: mukhdeep@gmail.com *

Abstract

The main purpose of studding computational intelligence as independent field is to understand and apply the principles that make intelligent behaviour possible weather it is from a natural or an artificial system. To accomplish this broad concept, it is important to understand different computational methods in which their popularity becoming increasing from time to time in the past few decades. The Paradigms of computational intelligence (CI) techniques have been successfully used in recent years to address various challenges of the real world which can't be solved by conventional techniques. The application areas are also numerous as some of them explained in the document. These techniques are inspired from fuzzy logic, evolutionary computing, learning theory and probabilistic methods. All these sub fields are different in nature and have different application areas and ability to solve problems. Nowadays, Techniques of Computational Intelligence has got great attention in solving problems in many fields of studies. In this document components of computational intelligence were explained by focusing on nature inspired algorithms. Comparative analysis of the techniques was made and shown that Swarm Intelligence based techniques has many advantages over the other techniques. The application of these techniques depends on the nature of the problem and experience of the researcher going to apply it. If they are properly handled these algorithms are wonder full to solve hard problems due to their stochastic nature.

Keywords **–**Computational intelligence, Nature Inspired Algorithms, Evolutionary Algorithms, Swarm based Algorithms

1. Introduction

In addition to biological inspirations that is a wide umbrella and very important for many researchers in the field of Computational Intelligence (CI) which covers only the main themes neural, fuzzy and evolutionary algorithms, other theories like the whole Bayesian foundations of learning, probabilistic and possibilist reasoning, many alternative approaches to handle uncertainty, kernel methods, search algorithms and many others are also under this field [1]. Generally, multi interdisciplinary and combinations of different theories and methods from other independent disciplines, including modern adaptive control, optimal control, learning theory, reinforcement learning, fuzzy logic, neural networks and evolutionary computation are

constituting Computational Intelligence [2]. What does computational intelligence study and where are the application areas are more meaningful in this newly shining computational field? Rather than giving specific definition. CI studies and becoming powerful for those problems there are no effective algorithms, either because it is not possible to formulate them or because they are NP-hard and thus not effective in real life applications. According to [1] computational intelligence is a branch of computer science studying problems for which there are no effective computational algorithms.

Computational intelligence was first proposed by Bezdek in 1992 [3] and defined it asa combination of soft computing and numerical processing. It was originated from emulating intelligent phenomenon in nature, computational intelligence centres on the study of adaptive mechanism to enable or facilitate intelligent behaviour in complex and changing environment. In the point of nature of view, it is a collection of nature-inspired computational techniques and approaches to address complex real-world problems to which mathematical or traditional modelling was failed due to certain reasons: This may be because of, too high complexity for mathematical reasoning, there might be some reservations and inconvenience during the process or the process might simply be stochastic in nature for computation by conventional method. As many real-life problems cannot be translated into binary language (unique values of 0 and 1) for computers to process it. Computational Intelligence therefore can give solutions for such problems.

The methods used in the field of computational intelligence are close to the human's way of reasoning, i.e. it uses inexact and incomplete knowledge and it is able to produce control actions in an adaptive way within a reasonable time. CI therefore uses a combination of five main complementary techniques:-The fuzzy logic mainly involved in the computer to understand vague, imprecise, uncertain, inexact, ambiguous or probabilistic knowledges as natural language, artificial neural networks which makes the system to learn experiential data by operating like the biological system, evolutionary computing, which is based on the process of natural evolution, learning theory and probabilistic methods which also helps dealing with uncertainty and imprecision[4].

Except those main principles, currently popular approaches include biologically inspired algorithms such as swarm intelligence and artificial immune systems, which can be seen as a part of evolutionary computation are playing a role in image processing, data mining, natural language processing and artificial intelligence. Computational Intelligence is thus a way of performing like human beings. Indeed, the characteristic of intelligence is usually attributed to humans [3]. More recently, many products and items also claim to be intelligent an attribute which is directly linked to the reasoning and decision making.

There are two types of machine intelligence: the artificial one that based on hard computing techniques and the computational one based on soft computing methods, which enable adaptation to many situations. Hard computing techniques work following binary logic

based on only two values (the Booleans true or false, 0 or 1) on which modern computers are based. One problem with this logic is that our natural language cannot always be translated easily into absolute terms of 0 and 1. Soft computing techniques, based on fuzzy logic can be useful here. Much closer to the way the human brain works by aggregating data to partial truths (Crisp/fuzzy systems), this logic is one of the main exclusive aspects of CI.

The five main principles of Computational Intelligence and it's applications

a) Fuzzy logic

As mentioned above, fuzzy logic is one of CI's main principles, which involves in measurements and process modelling made for real life's complex processes. It can face incompleteness and most importantly ignorance of data in a process model contrarily to Artificial Intelligence, which requires exact knowledge. Many researchers have been commented that measurements, process modelling and control can never be exact for real and complex processes. Also, there are uncertainties in the Fuzzy logic such as incompleteness, randomness and ignorance of data in the process modelling. In his work Zadeh introduced the concept of fuzzy logic to model human reasoning from imprecise and in complete information by giving definitions to vague terms and allowing construction of a rule base [5].

Fuzzy logic can accommodate the human experiential knowledge and give it an engineering flavour to model and control such ill-defined systems with nonlinearity and uncertainty. The fuzzy logic methodology usually deals with reasoning and inference on a higher level, such as semantic or linguistic [4].

The application of Fuzzy logic technology was a wide range of domains such as control, image processing and decision making. But it is also well introduced in the field of household appliances with washing machines, microwave ovens and in video camera, where it helps stabilizing the image while holding the camera unsteadily. In addition, the above-mentioned areas, we can apply it to medical diagnostics, foreign exchange trading and business strategy selection.

Fuzzy logic is mainly useful for approximate reasoning and it doesn't have learning abilities, a qualification much needed that human beings have. It enables human beings to improve themselves by learning from their previous mistakes.

b) Neural networks

Biological neurons are the fundamental building blocks of the brain. Neurons receive signals from neighbouring neurons through connections, process them in the cell body and transfer the results through a long fibre called an axon. An inhibiting unit at the end of the axon called the synapse, controls the signal between neurons. The axon behaves like a signal conducting device. An artificial neural network is an electrical analogue of the biological neural network. Neural networks originated from the work of Hebb in the 1940s and more recently the work of Hopfield, Rumellhart, Grossberg and Widrow in the 1980s has led to a resurgence of research interest in the field [6, 7]. Neural networks are biologically inspired, massively parallel

and distributed information-processing systems. Neural networks are characterized by computational power, fault tolerance, learning from experiential data and generalization capability and are essentially low-level computational algorithms that usually demonstrate good performance in processing numerical data. Learning was takes place in different forms in neural networks, such as supervised, unsupervised, competitive and reinforcement learning.

Regarding its applications, neural networks can be classified into five groups: data analysis and classification, associative memory, clustering generation of patterns and control. Generally, this method aims to analyse and classify medical data, proceed to face and fraud detection and most importantly deal with nonlinearities of a system in order to control it [8]. Furthermore, neural networks techniques share with the fuzzy logic ones the advantage of enabling data clustering.

c) Evolutionary computation

Based on the process of natural selection firstly introduced by Charles Robert Darwin, the evolutionary computation consists in capitalizing on the strength of natural evolution to bring up new artificial evolutionary methodologies. It also includes other areas such as evolution strategy and evolutionary algorithms which are seen as problem solvers. This principle's main applications cover areas such as optimization and multi-objective optimization, to which traditional mathematical one techniques aren't enough anymore to apply to a wide range of problems such as DNA Analysis, scheduling problems.

d) Learning theory

As far as the way of reasoning close to the human being, learning theory is one of the main approaches of computational intelligence. In psychology, learning is the process of bringing together cognitive, emotional and environmental effects and experiences to acquire enhance or change knowledge, skills, values and world views are discussed in literatures [9, 10]. Learning theories then helps understanding how these effects and experiences are processed, and then helps making predictions based on previous experience.

e) Probabilistic methods

Probability theory has been viewed as the methodology of choice for dealing with uncertainty and imprecision. The probabilistic method involves considering an appropriate probability space over a wider family of structures and proving that a sample point corresponding to the required structure has positive probability in this space. This method was discussed in the literature [11] and has made major contributions in areas of mathematics and computer science such as combinatorics, functional analysis, number theory, topology, group theory, combinatorial geometry and theoretical computer science. Probabilistic behaviour or stochasticity (randomness) is also sometimes listed as an attribute of intelligent systems. A complex nonlinear dynamic system very often shows chaotic behaviour, that is, chaotic phenomena are features of complex dynamical systems as mentioned in the literature [12]. Its aim was to evaluate the outcomes of a Computation Intelligent system, mostly defined

by randomness.[13] Therefore, probabilistic methods bring out the possible solutions to a reasoning problem, based on prior knowledge.

2. Application areas of Computational Intelligence

Nowadays, applications of Computational Intelligence rapidly increasing from time to time in the field of science mainly, computer science, engineering, data analysis and bio-medicine areas. Particularly its techniques have been successfully employed in a wide range of application areas, including decision support, generic clustering and classification, consumer electronic devices, stock market and other time-series prediction, combinatorial optimisation, medical, biomedical and bioinformatics problems. Although CI techniques are often inspired by nature or mimic nature in some way, CI applications are not restricted to solving problems from nature.

3. Nature Inspired Algorithms

Since the aim of this survey paper is to apply the algorithm derived from nature, it is more beneficiary to narrow the concept of computational intelligence to the sub field called nature inspired algorithms and discuss it in more detail.

Humans activity was always governed by nature due to there is certain mechanism in the nature to solve very complex Optimization problems in its own distinctive way. Real world problems around us are becoming more and more complex and at the same instance our mother nature is showing her experience to us how to solve these natural problems. Nature gives some of the logical and effective ways to find solution to these problems. Nature by itself acts as an Optimizer or Compromiser for solving the complex problems. Due to this imitation of processes running in real nature, the algorithms are named as "Nature Inspired Algorithms". The algorithms inspired from biological activity of human body with its working and the algorithms inspired from the working of groups of social agents like ants, bees and insects are the two classes of solving such Optimization Problems. As discussed in the literatures this area was newly emerging and needs great attention of researchers [14].

Problem solving methodologies in nature can be broadly categorized in to two branches: Classical or Exact methods (logical, mathematical programming) and Stochastic or Heuristics searching. Heuristic approach seems to be superior in solving hard and complex optimization problems, particularly where the traditional methods fail. Nature inspired algorithms are such heuristics that mimics /imitate the strategy of nature since many biological processes can be thought of as processes of constrained optimization. They make use of many random decisions which classifies them as a special class of randomized algorithms. To fix the structure of the bioinspired algorithms, a proper representation of problem must be chosen, check the quality of solution using a fitness function and defining proper operators so as to produce a new set of solutions [15].

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Optimization problems are commonly encountered mathematical problem in all engineering disciplines. As the reader can guess, it's literally meaning is finding the best possible or desirable solution among the given alternatives. Optimization problems are wide ranging and numerous, hence methods for solving these problems ought to be, an active research topic too. Optimization algorithms can be either deterministic or stochastic in nature. Former methods to solve optimization problems require enormous computational efforts, which tend to fail as the problem size means number of variables increases and/or nature of the problems become nonlinear. This is the motivation for employing bio inspired stochastic optimization algorithms as computationally efficient alternatives to deterministic approach. Meta-heuristics (the combination of Heuristic algorithms) are based on the iterative improvement of either a population of solutions (as in Evolutionary algorithms, Swarm based algorithms) or a single solution as (Tabu Search) and mostly employ randomization and local search to solve a given hard optimization problem [15, 14].

We roughly define hard optimization problems as problems that cannot be solved to optimality, or to any guaranteed bound, by any exact (deterministic) method within a reasonable time limit. These problems can be divided into several categories depending on whether they are continuous or discrete, constrained or unconstrained, mono or multi-objective, static or dynamic. In order to find satisfactory solutions for these problems, meta heuristics can be used. A meta heuristic is an algorithm designed to solve approximately a wide range of hard optimization problems without having a deep understand of each problem. Indeed, the Greek prefix ''meta'', present in the name, is used to indicate that these algorithms are ''higher level'' heuristics, in contrast with problem-specific heuristics. Metaheuristics are generally applied to problems for which there is no satisfactory problem-specific algorithm to solve them. They are widely used to solve complex problems in industry and services, in areas ranging from finance to production management and engineering. Almost all metaheuristics share the following characteristics: they are nature-inspired (based on some principles from nature); they make use of stochastic components (involving random variables); they do not use the gradient or Hessian matrix of the objective function; they have several parameters that need to be fitted to the problem at hand.

A vast literature exists on bio inspired approaches for solving an impressive array of problems and more recently, a number of studies have reported on the success of such techniques for solving difficult problems in all key areas of computer science. The two most predominant and successful classes or directions in nature inspired algorithms involves Evolutionary Algorithms and Swarm based Algorithms which are inspired by the natural evolution and collective behavior in animals respectively. Still this broad category classified in to different groups as highlighted next.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

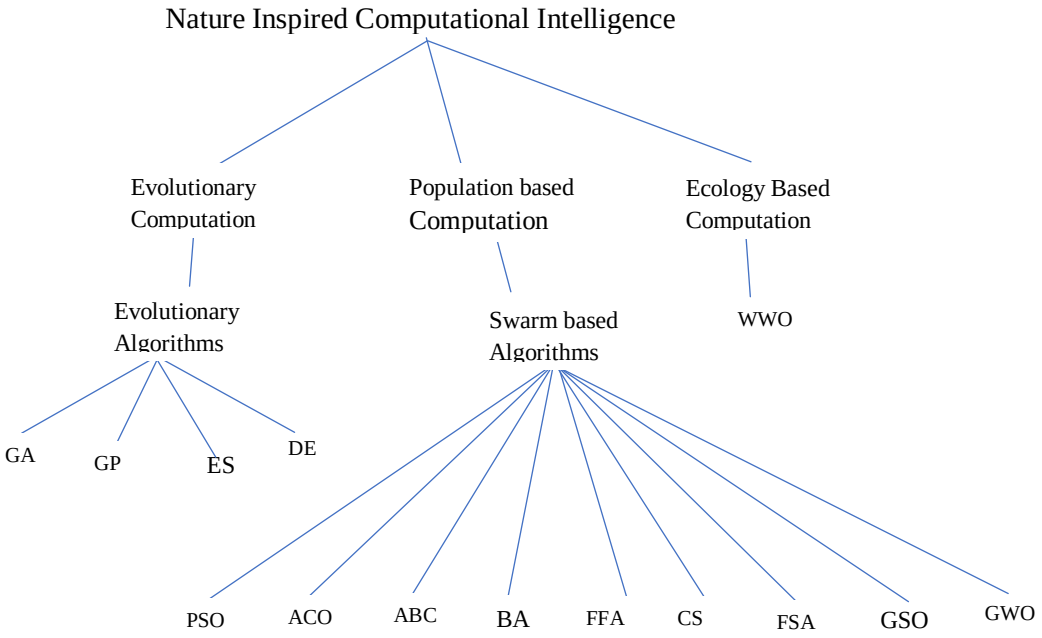

Fig 2. Taxonomy of Nature Inspired Computational Intelligence [15]

These classes of computational algorithms are grouped by their area of inspiration and the brief explanation of each are next with reference for detail.

3.1 Evolutionary Algorithms

Evolutionary algorithm is a part of Evolutionary computation which is a sub field of nature inspired algorithms. It refers to evolutionary computational models using randomness genetic inspired operations which involves selection, recombination, random variation and computation of individuals. It is general term for several computational techniques which represents power full search Optimization paradigm that influenced by biological mechanisms of evolution that are natural selection and genetics.

Evolutionary computation (EC) [16]is the core of artificial intelligence realm that aims at advancing from collective phenomena in adaptive populations of problem solvers utilizing the iterative progress of evolution which includes comprising growth, development, reproduction, selection, and survival as seen in a population under consideration for specific purpose. It is the most popular, classical and established algorithms among those nature inspired algorithms, which mimics the biological evolution of Charles Darwin theory of Natural Evolution. Over many stages of life, biological organisms develop according to the principles of natural selection

like "Survival of The Fittest" to attain some astounding form of accomplishment. Since it was highly effective in nature, it should be interesting to imitate natural evolution and try to procure a technique which may solve existing search and optimization problems. EAs employ this powerful design philosophy to find solutions to hard problems in the real world. EAs are non-deterministic algorithms or cost based optimization algorithms.

Imitation of biological activity of human body are viewed in different angles. These generate sub groups of Evolutionary Algorithms (EA) [15, 16]. The family groups of EAs comprises Genetic Algorithm (GA), Genetic Programming (GP), Differential Evolution and evolutionary strategy (ES). The members of the EA family share a great number of characteristics in common. They are all population-based stochastic search algorithms performing with best-to-survive criteria. The algorithm starts by creating an initial population of feasible solutions and continuous the process of iteration from generation to generation towards a best solution. In order to proceed successive iterations of the algorithm up to the desired solution, fitness-based selection will be made within the population of solutions. Better solutions are preferentially chosen for survival into the next generation of solutions.

3.2 Genetic Algorithms

Genetic algorithm is an evolutionary based stochastic optimization algorithm with a global search potential and developed by J. Holland, K.DE. Jong, D. Goldberg in 1970's [17]. It is among the most successful class of algorithms under EAs which are inspired by the evolutionary ideas of natural selection. Genetic algorithms are metaheuristic algorithms which mean that they generate from nature (based on some principles from physics, biology or ethnology); they make use of stochastic components (involving random variables); they do not use the gradient or Hessian matrix of the objective function; they have several parameters that need to be fitted to the problem at hand to produce good solutions of the time to optimize the problems.

The basic concept of GA is designed to simulate the processes in the natural system necessary for evolution, specifically as the reader can guess, GAs often imitates the principles first proposed by Charles Darwin of survival of the fittest. Since it was effectively implemented in the optimization of many nonlinear hard real-world problems, it is considered to be a great function optimizer. In GA, firstly initial set of population is selected from a random assemblage of solutions (Chromosome), for each chromosome evaluate the fitness values using an appropriate fitness function suitable for the problem. Then the other genetic operators such as crossover and mutation are very powerful in generating diverse solutions or search points, while elitism, adaptation and selection of the fittest help to ensure the proper convergence of genetic algorithms [18].

Parameter choices are also important in GA. But there are many parametric studies in the literature, and the overall literature of genetic algorithms is vast. In essence, the crossover should be more frequent with the highest probability, often above 0.7 to 0.95[19]. On the other hand,

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

mutation rate should be very low, because if the mutation rate is too high, the solutions generated are too diverse, and thus makes it difficult for the search process to converge properly. Therefore, mutation rate is typically 0.1 to 0.01.

Under this all process the best chromosomes are selected into the mating pool, where they undergo cross over and mutation thus giving new set of solutions(offspring). Each solution is depicted through a chromosome. Here these generated solutions are not exactly the desired answers. Genetic algorithms have many variants and often combined with other algorithms to form hybrid algorithms, and encode and decoding can be binary, real or even imaginary. Interested readers can refer to the recent books, for example, Goldberg [20] and other relevant books listed in the References. The three principal genetic operators in GA involve selection, crossover, and mutation.

GA is useful and efficient when:

- The search space is large, complex or poorly known.
- No mathematical analysis is available.
- Domain knowledge is scarce to encode to narrow the search space.
- For complex or loosely defined problems since it works by its own internal rules and
- Traditional search method fails.

Even though, GAs can rapidly locate good solutions, for difficult search spaces, it has some disadvantages:

1) GA may have a tendency to converge towards local optima rather than the global optimum of the problem if the fitness function is not defined properly.
2) Operating on dynamic data sets is difficult.
3) For specific optimization problems and given the same amount of computation time, simpler optimization algorithms may find better solutions than GAs.
4) Gas is not directly suitable for solving constraint optimization problems. Instead it was more power full for unconstrained and unbounded problems.

3.2.1 Basic Outline of Genetic Algorithm

1. *[start]* Create a population of n random candidate. (*i.e* suitable solutions for the problem).
2. [*Fitness*] Evaluate the fitness of $f(x)$ for each candidate.
3. [*New population*] Until the algorithm termination conditions are met, do the following (each iteration is called a generation).
   a) [*Selection*] Select two parent chromosomes from a population according to their fitness. (better the fitness has greater chance to be selected).
   b) [*Cross-over*] with Cross-over probability, Cross-over the parents to form new offspring (children). If no Cross-over was performed, the offspring is the exact copy of the parents.

c) [*Mutation*] with a mutation probability, mutate new offspring at each locus (position in chromosomes].
d) [*Accepting*] replace new offspring in the new population.

4. [*Replace*] Use generated population for a further run of the algorithm.
5. **[***Test*] If the end conditions are satisfied STOP and return the best population to the solution. If Not:
6. [*Loop*] go to step 2 to repeat the process by new population

3.2.2 Application areas of Genetic Algorithms

Genetic Algorithms was a population based evolutionary algorithms which is applicable in multi dimension areas of real world problems. Since the complexity of natural based problems are exponentially increasing as the number of variables and/or the searching space was become large the necessity and use of GA was also increasing. Some of the literatures apply GA in real world problems as follows.

Genetic Algorithms was applied in Parametric Design of Aircraft. Optimizing aircraft designs when the task is posed as that of optimizing a list of parameters, a genetic algorithm using real number representation, including the technique of generating a large number of initial population members effectively used to design [21]. It was also involved in dynamic anticipatory routing in circuit-switched telecommunications networks, by optimize the routing of telephone networks in order to minimize costs to. here genetic algorithm is a highly successful technique when the problem is complex, but hybridization of these algorithms can lead to better performance than using any of them in isolation. A genetic algorithm applied to robot trajectory generation, nonlinear dynamical systems, and models of international security, genetic synthesis of neural network architecture [22].

GA was also widely applied in strategy acquisition, genetic synthesis of neural network architecture, for conformational analysis of DNA, automated parameter tuning for sonar information processing a hybrid technique for engineering design optimization and the traveling salesman and sequence scheduling nonnegative matrix factorization with genetic algorithms for microarray analysis [23]. genetic algorithms also applicable for nonnegative matrix factorization of microarray analysis. nonnegative matrix factorization has proven to be a useful tool for the analysis of nonnegative multivariate data and we use a genetic algorithm for its optimization. the algorithm is applied to toy data to investigate its properties as well as to a microarray data set related to pseudo-xanthoma elasticum [24].

3.2.3 Limitations of Genetic Algorithms

Like any technique, which has merit and demerit GAs also suffer from a few limitations when applied to solve the real-world problems. These challenges include

- Gas is not suited for all problems, especially problems which are simple and for which derivative information is available. It means, for those problems in which their gradients and Hessian matrix are available defining GA is economical and sometimes fails to get an exact solution as determined method.
- Fitness value is calculated repeatedly for each generation, which might be computationally expensive for some problems.
- Being stochastic, there are no guarantees on the optimality or the quality of the solution.
- If not implemented properly, the GA may not converge to the optimal solution. Means that the method is highly dependent on the experience the researcher.

Generally, by controlling such constraints, Genetic Algorithms are power full and have the ability to deliver a good-enough and fast-enough solution. This makes genetic algorithms attractive for use in solving optimization problems

3.2.4 Genetic Programming

Genetic Programming (GP) was proposed by John Koza in 1992[25]. It is a representation of programming which adopts the concepts of natural evolution to solve a complicated problem. Genetic programming can be seen as a new version of the genetic algorithm. The main difference between Genetic Programs and Genetic Algorithm is in terms of representation of the solution.GP works with an indirect encoding of a potential solution (in the form of a tree), in which search is involved in to the solution directly, and a solution could be a computer program. The other fundamental difference is in the variable-length representation adopted by GP in contrast with the fixed length encoding in GA. It is an automatic process for fabricating a working computer program from a high level complex problem statement. GP reach to this point of automatic programming by genetically procreating a population of computer programs using the postulates of Darwinian natural selection and nature inspired operations.

Genetic Programming works in the following ways:

1. First arbitrarily create an initial set of population of independent computer programs made of functions and terminals.
2. Execute each program in the population and calculate fitness value for each program using a criterion decided in starting.
3. Select one or two independent program(s) from the population with using a probabilistic approach for fitness to take part in the genetic operations.
4. Formulate new independent program(s) for the population by applying the following genetic operations:
   - *Reproduction*: Replicate the selected independent programs to the new population.
   - *Crossover*: The next new construction of offspring program(s) for the new population

by recombining will be chosen arbitrarily from the selected programs.

- *Mutation*: Construct a single new offspring program for the new population by Choosing part of arbitrarily mutating of an arbitrarily selected program.

3.2.5 Application areas of Genetic Programming

In the field of artificial intelligence, genetic programming (GP) is a technique whereby computer programs are encoded as a set of genes that are then improved by using another evolutionary algorithm usually Genetic Algorithm. It is mostly applicable where the space of solutions consists of computer programs. The results are computer programs able to perform well in a predefined task. Hence, GP has a lot of application areas in the real-world problems.

Genetic Programming has been developed and extensively applied for analysis of molecular data to classify cancer subtypes and characterize the mechanisms of cancer pathogenesis and development [26]. Genetic programming (GP) is a flexible and powerful evolutionary technique for classification real problems such as credit scoring, bankruptcy prediction, medical diagnosis, pattern recognition, text categorization, software quality assessment and many more [27].

3.2.6 Limitations of Genetic Programming

Despite the advantages of GP, there are several limitations in its use. The first is that genetic programming is not in the category of an optimization algorithm in the sense of finding the global solution to a problem. During search for, there may be in fact be the global optimum, since genetic programming is a stochastic method that is driven only by a relative measure of fitness (i.e., how much fitter one individual program is to another), there is no way to tell whether a given solution is the best possible. Instead, the results must be assessed on the basis of what is needed and what is possible given the available data.

The other drawback, since we are often dealing with situations where the number of inputs is significantly large than the number of samples, the driving power that allows GP to find solutions makes it very easy to find solutions that are overfit to the training data and will not perform well on unseen samples. For this reason, we often work to find the most ungenerous solution that meets our halting criteria. This constrains the GP search process to look for the most general solutions within a set of data, which, based on Occam's razor, has a greater chance of being a general-purpose solution. We also typically review the solutions for internal consistency in how features are used among the population of solutions (e.g., whether a gene is consistently up-regulated in multiple rules predicting cancer recurrence) and, if possible, that at least some of the features identified make sense based on the published literature.

GP is a computationally intensive process that can require tens or hundreds of thousands of programs to be tested in searching for a desired solution. Even though the basic algorithm is highly parallel by its nature, and can effectively use parallel and grid computing techniques, it may still require more computing power than is readily available [27].

3.3 Evolution Strategies

Evolution Strategies was proposed at the beginning by three students Bienert, Reichenberg, Schwefel at the Technical University in Berlin in 1964[28] in an effort to robotically optimize an aerodynamic design problem. Evolutionary strategies belong to the domain of evolutionary algorithms that solve optimization problems by the concept of iterative procedure. It was a global optimization algorithm inspired from nature by the theory of adaptation and evolution by means of natural selection. Specifically, this evolutionary technique is inspired by macro-level or the species-level process of evolution like phenotype, hereditary and variation to the contrary of genetic algorithm which deals with micro or genomic level (genome, chromosomes, genes, alleles). A very important feature of ES is that, the utilization of self-adaptive mechanisms for controlling the application of mutation. These mechanisms are aimed at optimizing the progress of the search by evolving not only the solutions for the problem being considered, but also some parameters for mutating these solutions. Some most common Selection and Sampling schemes in ES are as follows:

(1+1)-ES: Evolutionary strategy works by a simple selection mechanism in which it creates one real-valued vector of object variables from its parent by applying mutation with an identical standard deviation to each object variable. Then, the individuals obtained through this system was evaluated and compared to its parent, and the better survives to become a parent of the next generation, while the other is discarded.

(μ +λ)-ES: Here the algorithm use μ parents to selected from the current generation and generate λ offspring, through some recombination and/or mutation operators was followed. Out of the union of parents and offspring (μ + λ), the best μ kept for next generation of the population. It inherently highly integrates with elitism.

(μ, λ)-ES: In this scheme the algorithms use μ parents to select from the current generation and used to generate λ offspring (with λ >= μ) and only the best μ offspring individuals form the next generation discarding the parents completely. This does not incorporate elitism.

3.3.1 Differential Evolutions

Differential Evolution (DE) is another paradigm in the Evolutionary Algorithm which was first proposed by Storn and Price in 1995[30] for global optimization over continuous search space. It is a population based stochastic search technique which is similar to GA in that

populations of individuals are used to search a problem by iteratively trying to improve a candidate solution with respect to a given measure of quality. The main difference of DE from GA is that, in DE arithmetic combinations of individuals is given by mutation where as in GAs, mutation is the result of small perturbations to the genes of an individual.

The mutation operator of DE considers exploration as evolution process at the beginning and as evolution progresses, the mutation operator favours exploitation. Hence, based on the stage of the evolutionary process DE automatically adapts the mutation increments to the best value. Mutation in DE is therefore not based on a predefined probability density function. Mutant vector is created from three different randomly selected targeted vectors. New trial vector is obtained from the target vector and the mutant vector based on the crossover probability.

3.3.2 Application areas of Differential Evolution

As discussed above DE was one central part of evolutionary algorithms and applicable in real Optimization problems. Especially those NP hard optimization problems are tackled by the help of this algorithm. Differential Evolution (DE), which was formulated as a stochastic parallel direct search method, which have been used concepts from the broad class of evolutionary algorithms, but requires few easily chosen control parameters to find the optimal solutions for nondifferentiable, nonlinear and multimodal objective functions and also single and multi-objectives function optimization, neural network training, clustering, and real life DNA microarray problems are where this technique was applicable[29].

3.4 Swarm intelligence

Swarm Intelligence (SI) was proposed by Kennedy and Eberhardt in 2001[31], which is a recent and newly developing paradigm in nature inspired computing for implementing adaptive systems. Even though it was nature inspired it alien in different direction with Evolutionary Algorithms. While Evolutionary Algorithms are based on genetic adaptation of biological organisms, Swarm intelligence (SI) is based on collective social behavior of groups organisms. As per definitions in literature, Swarm Intelligent incorporates the implementation of collective intelligence of groups of simple agents that are based on the behavior of real world insect and swarms, as a problem-solving tool. The word swarm comes from the irregular movements of the particles in the natural space.

SI was discussed as the collaborative movement of a group of animals, especially birds and insects such as ants, bees and termites, that are each following very basic rules but when seen in the field of computer science, swarm intelligence is a simulated way to problem solving using algorithms formed on the concept of self-managed collective behavior of social insects. Bonabeau*et al.*, 1999 [10] gave a simple definition for better understanding of swarm intelligence *i.e.*, "the popular way of simple and common intelligence of social agents". Swarm

intelligence is an emerging new domain that visualizes intelligence as a method of communication between independent agents.

Swarm intelligence (SI) is the characteristics of collective, emerging behaviour of multiple, interacting agents who follow some simple rules which can accomplish very trajectory work in group. While each agent may be considered as unintelligent which can't do nothing, the whole system of multiple agents may show some self-organization behaviour and thus can behave like some sort of collective intelligence. Swarm-intelligence systems in nature become base and source of inspiration for many algorithms have been involving in computational intelligence nowadays.

Several optimization algorithms inspired by the natural law of swarming behaviour in nature are proposed. Particle Swarm Optimization (PSO), Ant colony optimization (ACO), Firefly Algorithm (FA), Bat Algorithms (BA), Bee Colony Optimization (BCO), Fish swarming Algorithm (FSA), Cuckoo Search (CS) and Grey Wolf Optimization (GWO) are some of popular examples to this effect

Swarm Intelligences are: -

- Collective system that have the ability of accomplishing very difficult tasks in dynamic and varied environments without any external guidance or control and with no central coordination.
- Achieving a collective performance which could not normally be achieved by an individual acting alone.
- Constituting a natural model particularly suited to distributed problem solving.

The characteristic of swarm intelligence system has been given as the following: -

- The swarm is made up of a composition of many individuals/ Organisms.
- the individual organizes involved in to the system are relatively homogeneous (i.e., they are either all identical or they belong to a few typologies).
- all interactions between each of the individuals are based on simple adaptive behavioural rules that exploit only local information that the individuals exchange directly or via the environment.
- the overall behaviour of the system obtained from the interactions of individuals with each other and with their environment, that is, the group behaviour self-organizes.

The most interesting property of a swarm intelligence system is that, its ability to act in a coordinated way without the presence of a coordinator or of an external controller among the swarms. A lot of interesting movements have been observed in nature of swarms that perform some collective behaviour without any individual controlling the group, or being aware of the overall group behaviour. Nevertheless, the lack of individuals in charge of the group, the swarm

as a whole can show an intelligent behaviour. This is the result of the interaction of spatially neighbouring individuals that act on the basis of simple uncontrolled rules.

Most of the time, the behaviour of each individual of the swarm is described in probabilistic terms: Each individual has a stochastic behaviour that depends on his local perception of the neighbourhood individual. Because of the above properties, it is possible to design an algorithm of swarm intelligence system that are scalable, parallel, and fault tolerant.

- Scalability means that the whole system can keep its function while increasing or decreasing its size without the need to redefine the way its parts interact. Because the swarm intelligence system interactions involve only neighbouring individuals, the number of interactions tends not to grow with the overall number of individuals in the swarm: each individual's behaviour is only loosely influenced by the whole swarm dimension. In artificial systems, scalability is important because a scalable system can increase its performance by simply increasing its size, without the need for any reprogramming.
- Parallel action is possible in swarm intelligence systems because each individual in the composing the swarm can perform different actions in different places at the same time. In artificial systems, parallel action has great value because it can help to make the system more flexible, that is, capable to self-organize in teams that take care simultaneously of different aspects of a complex task.
- Fault tolerance is also an inherent property of swarm intelligence systems due to the decentralized, self-managed and self-organized nature of their control structures. Because the system is composed of many homogenous individuals and none of them is in charge of controlling the overall system behaviour, a failing individual can be easily dismissed and substituted by another one that is fully functioning.

3.4.1 Particle Swarm Optimization

Particle swarm optimization (PSO) [32] is inspired by nature and a computational intelligence oriented, stochastic, population-based global optimization technique developed in 1995 by Eberhardt and Kennedy based on the behaving aspects of birds flocking or fish schooling. In PSO, the term particles indicate to population members which are mass-less and volume-less (or with an arbitrarily small mass or volume) and are subject to velocities and accelerations towards a better mode of behavior. PSO adopts the nature's general behavior, rapidly changing movements and interaction among social agents such as birds or fishes. It is based on the simulation of the social behavior of birds flocking, fish schooling, and animals herding to let them adjust to their environment, search rich sources of food, and protecting from other predatory animals by using information sharing and social cognitive intelligence.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

PSO combines the individuals experience with experiences which are learned when working in group to derive simple model that can deal with complex problems effectively. This algorithm uses a number of agents (particles) that constitute a swarm moving around in the search looking for the best solution. Each particle in the search space changes its flying position according to its own personal flying experiences as well as the flying experiences of other particles in the group. Each particle keeps an eye of its own coordinates in the solution space which are correlated with its best fitness value, attained by it so far. This value is called personal best, p best. Another important best value for the fitness is that the one that is considered by the PSO considered as the best fitness value obtained so far by any particle in the close range of that particle. This value is called global best or g best. The main idea of PSO lies in speeding each particle toward its p best and the g best locations with an arbitrarily weighted speeding factor at each time step. Every particle seeks to alter its coordinates based on some equation and parameters [32]: The current position, current velocities, distance between the current position and p best, the distance between the current position and g best. The alteration of the particle's velocity and position can be mathematically described using following equations:

$$v_{k+1}^{i} = v_{k}^{i} + c_1 r_1 \left(p_{k}^{i} - x_{k}^{i}\right) + c_2 r_2 \left(p_{k}^{g} - x_{k}^{i}\right) \quad (1)$$

$$x_{k+1}^{i} = x_{k}^{i} + v_{k+1}^{i} \quad (2)$$

Where, x_k^i represents Particle position, v_k^i Particle velocity, p_k^i represents Best "remembered" position, $c_{1,}c_2$ represents cognitive and social parameters and $r_{1,}r_2$ are random numbers between 0 and 1

Steps to be followed in implementing PSO algorithm can be briefed as below: -

1) Initialize the random swarm by assigning a position in the problem space to each individual particle.
2) Calculate the value of fitness function for each of individual particle.
3) For each individual particle, compare the particles fitness value obtained above with its p bests. If the current value is better than the p best value, then set this value as the p bests and the current particles position, x_i, as p_i.
4) Identify the particle that has the best fitness value. The value of its fitness function is identified as guest and its position as p_g.
5) Update the obtained velocities and positions of all the particles using (1) and (2).
6) Repeat steps 2 to 5 until the designed stopping criterion is met (e.g., maximum number of iterations or a sufficiently good fitness value).

The advantages that using PSO has over Genetic Algorithm:

(a) PSO is easier to implement, has short steps and there are fewer parameters to adjust.
(b) PSO has a more effective memory capability than GA at the time of computation.
(c) PSO is preferable over the others computational techniques in maintaining the diversity of the swarm, since all the particles in the swarm uses the information related to the most successful particle in order to improve themselves, whereas in Genetic algorithm, the bad solutions are discarded totally and only the new ones are saved; i.e. in GA the population evolve around a subset of the best individuals. Meaning the fittest individuals are considered to continuing the process.

There are many similarities between the PSO and EAs. Both of them initialize solutions and update generations, while the PSO has no evolution operators as does the latter. In a PSO, particles try to reach the optimum by following the current global optimum instead of using evolutionary operators, such as mutation and crossover. It is claimed that the PSO, in addition to continuous functions, has been showing stability and convergence in a multidimensional complex space also.

Application Areas of PSO

Particle Swarm Optimization can be used to solve a number of scheduling problems, detection of road accidents [33]. Even though data clustering problems were solved using computational intelligence information technologies, with the help of PSO it can be solved more efficiently. For their construction of cell dimension in breast cancer and to locate its position, a technique using finite - difference frequency domain with PSO Fuzzy C means clustering technique in combination with PSO is used for classification approach and for detection of micro calcification in digital mammograms and also for enhancing the segmented mammogram images [34]. It can also be used as a hybrid approach for detecting and classifying the micro calcification in mammogram images. A combination of genetic and particle swarm optimization is used to optimize the feature sets that improves the classification accuracy in digital mammograms images. PSO, with the help of k-Neural Networks, perform shape based diagnosis scheme for feature selection and classification can be easily done, for the estimation of permittivity of the tissue layers in breast for detecting cancer [35]. A heuristic PSO algorithm has been proposed for solving the span problems for job on unrelated parallel machines. A Fuzzy Support Vector Machine in hybrid with PSO is modelled for tacking imbalanced classification problems in mammograms. This algorithm can also be used to classify the face emotions through eye and lip features [36].

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

3.4.2 Ant Colony Optimization

Ant Colony Optimization (ACO) is a population based meta heuristic algorithm which takes its inspiration from the real world of ant colonies by the searching behaviour of their searching food to solve optimization problems. It was first introduced by Marco Dorigo in 1992 [37].

For finding food, ants start out from their colony and move randomly in all directions individually. Since individual ants are unable to communicate or work efficiently, hence they work in societies and communicates with each other through a chemical substance known as pheromone. Once an ant find food, it returns to colony and leave a pheromone along the pathso that other ant can follow the route while searching for the food. Other ants of the swarm can sense pheromone trails and move on the same path. The interesting point what we can observe from this natural activity is that, how often the path visit by ants is determined by the concentration of pheromone along the path. Since pheromone will naturally evaporate over time, the length of the path is also a factor. Therefore, under these conditions, a shorter and best path will be chosen because ants moving on that path keep adding pheromone to it which makes the concentration strong enough against evaporation. As a result, the shortest and best path from colony to food emerges. This type of interaction between social agents is known as stigmergy. Stigmergy is a mechanism of indirect coordination between agents or actions. The law of stigmergy says that the indications left in the environment of current action simulate further the next subsequent actions [37]. This whole process of simulation in last creates a systematic way of performing some activity.

The main motive behind this algorithm is to solve problem of searching an optimal path using weighted graph technique, also known as construction graph. Artificial ants are used to search best paths in the graph. Real ants deposit pheromone trail on their forward and return journey while artificial ants deposit pheromone trail only after solution has been constructed.

To optimizing the travelling salesman problem, ant system has been utilized, feature extraction from the mammogram images can be easily done by using the nature of ant colonies while searching for food.

ACO algorithm works in this way: -

1. *Initialization:* Initialize ACO parameters (*e.g.*, no. of artificial ants) and pheromone trails.
2. *Construct Ant Solutions***:** A set of some artificial ants' construct solutions from elements of a finite set of available solution components.
3. *Daemon Actions***:** When solution set has been created, before increasing or decreasing pheromone values, some particular actions are to be taken according to the problem given.

These actions are known as Daemon Actions. These actions vary problem to problem. The Daemon Actions are optional.

4. *Update Pheromone*: Update pheromone is the mechanism of increasing or decreasing the pheromone values along the path. The pheromone values are increased for good solutions and decreased for bad solutions. There is another procedure which is executed inside update pheromone action which is evaporation. The pheromone values are evaporated so that bad solutions which were learned earlier can be forgotten.

Application areas of ant colony

As the same process of searching for food by ants mentioned above, the algorithm inspired by ant colony is applied in a number of applications for finding the optimal solutions. Originally, ant colony optimization was applied to travelling salesman problem and then applied later to various hard optimization problems [38]. It is also used to solve various kinds of distributed control problems and difficult optimization problems, for segmenting the MRI brain images for detection of tumour and for solving the discrete optimization problems. It can also be implemented for extraction of suspicious regions using an approach which is asymmetric. This algorithm can also be utilized for search procedure and for implementing feature subset selections [39].

3.4.3 Bat Algorithm

Bat algorithm is a relatively newly emerging metaheuristic, developed by Xin-She Yang in 2010. [40, 41, 42]. It was inspired by the echolocation behaviour of microbats. Microbats use a type of sonar, called echolocation, to detect prey, avoid obstacles in front of them and locate their roosting crevices in the dark. These bats naturally release a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. The direction of their pulses varies in properties and can be correlated with their hunting strategies, depending on the species. Almost all bats use short, frequency-modulated signals to sweep through about an octave, while others more often use constant-frequency signals for echolocation. Their signal bandwidth varies depends on the species and often according to increase by using more harmonics.

Inside the bat algorithm, it uses three main idealized rules: -

1. All bats use echolocation to sense distance and they also know the difference between food/prey and background barriers in some magical way gifted by nature.
2. Bats fly randomly with velocity $v_i$at position $x_i$with a fixed frequencyf_{min}, varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0,1]$, depending on the proximity of their target.
3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}.

BA has been extended to multi objective bat algorithm (MOBA) by [43], and preliminary results suggested that it is very efficient and more power full to solve problems.

3.4.4Firefly Algorithm

Firefly Algorithm (FA) was developed by Xin-She Yang at Cambridge University [44, 45, 46, 47], which was based on the flashing patterns and behaviour of fireflies. In their movement, each firefly will be attracted to brighter ones, while at the same time, it explores and searches for prey randomly. In addition, the brightness of a firefly is calculated by the landscape of the objective function designed the particular problem. The movement of a firefly *i* is attracted to another more attractive (brighter) firefly *j* is determined by

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2} (x_j^t - x_i^t) + \alpha^t \varepsilon_i^t \quad (1)$$

where the second term is due to the attraction. The third term is randomization with α^tbeing the randomization parameter, and $\varepsilon_i^t$is a vector of random numbers drawn from a Gaussian distribution or uniform distribution. Here is $\beta_0 \in [0,1]$ is the attractiveness at $r = 0$, and $r_{ij} = \left\| x_i^t - x_j^t \right\|$is the Cartesian distance. For other problems such as scheduling, any measure that can effectively characterize the quantities of interest in the optimization problem can be used as the 'distance' *r*. For most implementations, we can take $\beta_0 = 1, \alpha = O(1)$ and $\gamma = O(1)$.
Ideally, the randomization parameter *t* should be monotonically reduced gradually during iterations. A simple scheme is to use.

$$\alpha_t = \alpha_0 \delta^t \ , \delta \in (0,1) \quad (2)$$

where α_0 is the initial randomness, while $\delta$is a randomness reduction factor similar to that used in a cooling schedule in simulated annealing. It is worth pointing out that (4) is essentially a random walk biased towards the brighter fireflies. If $\beta_0 = 0$, it becomes a simple random walk. Furthermore, the randomization term can easily be extended to other distributions such as Lévy flights. A basic implementation can be obtained from this link.1 High nonlinear and non-convex global optimization problems can be solved using firefly algorithm efficiently [48]

Application areas of fire Fly

Firefly algorithm is also a swarm based algorithm that was designed to evaluate the exploratory behaviour of fireflies for finding the optimum value of fitness functions under the given set of constraints. This algorithm imitates the firefly interaction in natural phenomena, least computation time in compressing the digital images. It produces consistent and more

accurate performance in terms of time and optimality for feature selection [49]. Fire Fly algorithm can also be efficiently used in biometric technology for personal authentication and identification like dorsal hand vein recognition that gives better results when compared with other algorithms. For solving highly nonlinear, multimodal design firefly algorithm provides the best efficiency solutions [50]. It also has been applied for the optimum design of antenna and shows better performance than other artificial design algorithms. NP-Hard problems, multi objective load dispatch problems, scheduling problems etc. can be easily solved, has better performance and efficiency when solved with firefly algorithm and also firefly algorithm solve the scheduling problems in permutation flow shops and travelling salesman problem in a very promising way. By optimizing the network parameter, we can use the firefly algorithm to improve the performance of local linear wavelet neural network for classifying the breast cancer [34].

Modified firefly algorithm proves better results while detecting the edge strength and optimal threshold selection in images and also in enhancing the image and also an efficient system that is used to distinguish a defective eye from a normal eye [50] . As a pre-processing step in offline signature verification system and also for feature extraction. This algorithm can also be used to solve maximum weighted satisfaction problems and for developing the more optimization algorithm.

3.4.5 Artificial Bee Colony Algorithm (ABC)

The concept of Artificial bee colony (ABC) was first introduced by Karaboga in 2005[51] is the most popular algorithm based on bee colony optimization. In ABC the bee colonies foraging behavior is imitated. As the ants search their food using pheromones discussed above, similar food collecting behaviors are found in honey bees. Instead of pheromones, the algorithm imitates the foraging behavior of honey bees. The first step in this process is to sending bees in different directions to search for. The bees use a technique known as waggle dance total other bees about the food source and good quality food including its location to their colony. This dance tells three information to other bees in the colony. This three information are distance of food source from colony, the direction in which all other bees have to go and last the quality of food source. The bees are attracted to that bee only which has brought information about the best quality food source recently found. This process gives the best food source for Bee Colony naturally.

In the ABC algorithm, there are three groups of bees which exists in the colony, namely, *employed bees, onlookers and scouts.* A bee under the category of guarding on the dance area for making a decision to choose a food source is called onlooker and one going to the food source visited by it before is named employed bee. There is also other kind of bee, called scout bee. This kind of bee carries out random search for discovering new sources of food. The position of a food source obtained by scout bee represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

(fitness) of the associated solution. A swarm of virtual bees are generated and started to move randomly in directions of the search space. Bees are highly cooperative when they find some target nectar and the solution of the problem is obtained from the intensity of these bee interactions.

Some randomly distributed initial population solutions$(x_i = 1,2 \dots D)$ is being dispreads over the D dimensional problem space. The work of an employed bee is always to updates the existing position (solution) in her memory based on the local information (visual information) it has before and tests the nectar amount (fitness value) of the newly found source (new solution). If the nectar amount of the new one is higher than that of the previous one, the bee memorizes and give attention to the new position and forgets the old source. After all employed bees completed their search process for new source; they share the nectar information of the food sources and their position information with the onlooker bees on the dance area.

In the next phase Reproduction, based on the probability value associated with the food source, $\boldsymbol{Pi}$, the artificial onlooker bee chooses a food source

$$P_i = \frac{Fit}{\sum_{n=1}^{N} Fit_n} \quad (1)$$

Where, N is the number of food sources (that is the number of employed bees), $fit_i$is the fitness value of the solution$i$ which is proportional to the nectar amount of the food source in the position$i$.

In the last phase, Replacement of bee and Selection*, when* a position cannot be improved further through a predetermined number of cycles, in this case that food source is assumed to be abandoned. The value found for the preassigned number of cycles is an important and used to control parameter of the ABC algorithm, which is called "$limit$" for abandonment. After candidates of each source position is produced and then evaluated by the artificial bee, its achievement is compared with that of its old one. If the newly found food has an equal or better nectar than the old source, it replaces the old one in the memory. Otherwise, the old one is retained in the memory.

ABC algorithms local search accomplishment depends on neighborhood search and greedy selection mechanisms performed by the assigned employed bees and onlooker bees. The performance of the global search of this algorithm depends on random search process performed by scouts and neighbor solution production mechanism performed by employed and onlooker bees.

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Application areas of Artificial Bee Colony Algorithm (ABC)

ABC algorithms has been employed to solve graph colouring problems [52], for finding clusters of an image [53]. With ABC algorithm a technique was proposed that is used for removing Doppler noise in the aortic valve and also for finding filter coefficients, by using heuristic ABC algorithm a new approach has been developed that is used for matching templates in digital images [52]. Also, ABC is used for solving course scheduling problems, job shop scheduling problems, Steiner tree problem and for diagnosing the breast cancer.

3.4.6 Cuckoo search

Cuckoo search (CS) is an evolutionary optimization algorithm that is inspired by Yang and Deb (2009) and it has attracted great attention due to its promising efficiency in solving many Optimization problems and real-world applications. The theory of cuckoo search was inspired by the species of bird called the cuckoo. Cuckoos are one of very attractive birds' species, not only because of the unique and beautiful sounds they can produce, but also because of their hostile reproduction strategy, by which mature cuckoos lay their eggs in the nests of other host birds or species. This mechanism was known as obligate brood parasitism. Each egg in a host's nest represents a solution, and a cuckoo egg represents a new solution of the population. Basically, the ground rule of this algorithm is the specific egg laying and breeding of cuckoos themselves. At the time, when a host bird discovers the eggs in her nest are not its own, it will it will take an aggressive action as either throw these alien eggs away or simply forget its nest and build a new nest elsewhere [54].

In general, the cuckoo eggs usually hatch earlier than the host eggs. Some of these eggs also can mimic or look similar to the host bird's nest and have the opportunity to grow up and get more feeding than the host birds. Then, if these eggs are not recognized by host birds, they will grow and become mature cuckoos. Cuckoos those are used in this modelling exist in two forms, which are mature cuckoo sand eggs. After remaining eggs grow and turn in to mature cuckoos, they will form a group or society. Environmental features and the immigration of societies or groups of cuckoos hopefully lead them to converge and find the best environment for breeding and reproduction. The CS algorithm finds the global maximum of objective functions through the best environment for breeding and reproduction [55].

Some species of cuckoos like the Ani and Guira shared other birds' nests to lay their eggs, though they may remove others' hosts eggs to increase the hatching probability of their own eggs. In the cuckoo search algorithm, we may have three basic types of brood parasitism: These are: -intraspecific brood parasitism, cooperative breeding, and nest takeover. Some host birds can engage direct conflict with the intruding cuckoos. If a host bird realizes that the eggs are not its own, it will either clean the nest by removing these alien eggs away or simply abandon its nest and build a new nest elsewhere. There are some cuckoo species in this process such that the new world brood-parasitic Tapera have evolved in such a way that female parasitic cuckoos

are often very specialized in the mimicry in colour and pattern of the eggs of a few chosen host species. This reduces the probability of their eggs being abandoned and thus increases their reproductively [56].

Cuckoo Search Methodology

Cuckoo Search is basically based on three idealized important rules.

a. Each cuckoo lays one egg at a time and dumps its egg in a randomly chosen other bird nest.

b. The best nest with high-quality eggs will carry over to the next generation.

c. There are number of fixed available hosts' nests and the egg laid by a cuckoo is discovered by the host bird with a probability $p_a \varepsilon [0, 1]$.

Depending on the above mentioned three rules of cuckoos searching process, the possibility is that the host bird can either remove the egg away or leave the nest and build a completely new nest. For simplicity, this last assumption can be approximated by the fraction p_a of the n nests that are replaced by new nests (with new random solutions).

For a maximization problem, the quality or fitness of a solution can simply be proportional to the objective function. Other forms of fitness can be defined in a similar way to the fitness function in GA. Based on the three rules given above, the basic steps of the Cuckoo Searches can be summarized as the pseudo code as follows: When generating new solutions $x^{(t+1)}$ for, say cuckoo i, a $Le'vy$ flight is perform.

$$x_i^{(t+1)} = x_i^t + k \oplus Le'vy(\lambda) \quad (1)$$

where $k > 0$ is the step size which should be related to the scales of the problem of interest. In most cases, we can use $k = 0$ (9). The product $\oplus$ means entry-wise multiplications. Lévy flights essentially provide a random walk, while their random steps are drawn from a Lévy distribution for large steps

$$Lévy \sim u = t^{-\lambda}, (1 < \lambda \leq 3) \quad (2)$$

which has an infinite variance with an infinite mean. Here, the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail

CS is widely applied in engineering, pattern recognition, job scheduling, networking, Object-Oriented software (software testing) and data fusion in wireless sensor networks.

3.4.7 Fish Swarm Algorithm

The fish swarm algorithm (FSA) is a newly proposed swarm based intelligent evolutionary computation technique proposed by Li et al. in 2002 [57] which is inspired by the natural schooling behaviour of the fish. FSA is come up with a strong ability to avoid local minimums of the problems in order to achieve global optimization.

In the FS Algorithm, a fish is represented by its N-dimensional position $X_i = (x_1, x_2, \ldots, x_k, \ldots, x_N)$, and food satisfaction for each fish is represented as FS_i. The relationship between two fish is denoted by their Euclidean distance $d_{ij} = ||X_i - X_j||$ where, X represents the $i^{th} or j^{th}$ position of each particle in the swarm.

This algorithm, follow the three typical behaviours of fish schooling, given as: - searching for food, swarming in response to a threat, and trying to increase the probability of achieving a successful result as explained next.

***Searching for food*:** - is a random search adopted by fish in the search space through arbitrary direction for searching food, with a tendency towards food concentration. The objective of this search is to minimize FS (food satisfaction).

***Swarming*:** -Fishes are moving with the aims to satisfying food intake needs, entertaining group members and attracting new swarm members to the food source. A fish, whose location is at X_i has neighbours within its visual locationX_C identifies the centre position of those neighbours and issued to describe the attributes of the entire neighbouring swarm. If the swarm central position has greater concentration of food than is available at the fish 's current position $X_i (i.eX_C > X_i)$, and if the swarm (X_c) is not overly crowded, then the fish will move from X_i to next X_{i+1}, toward X_C .

***Following*:** - characteristics of fish shows, when a fish locates food then the neighbouring individuals follow. Within a fish 's visual position, certain fish will be perceived as finding a greater amount of food than others, and this fish will naturally try to follow the best one (X_{min}) in order to increase satisfaction for food ($i.e$ gain relatively more food which means $FS_{min} < FS_i$and less crowding. The Three import ant parameters involved in FSA are includes visual distance (visual), maximum step length (step), and a crowd factor. FSA effectiveness seems primarily influenced by the former two (visual and step).

3.4.8 Glow worm Swarm Optimization

Glow worm swarm optimization (GSO) algorithm is the one of the newly derived nature inspired metaheuristics for optimization problems proposed by Krishnan and Ghose in 2005[58]. This algorithm was a derivative-free, meta-heuristic algorithm, mimicking the glow behaviour of glow worms and have some common features with ant colony optimization (ACO) and with particle swarm optimization (PSO) with several significant differences. The individual agents in

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

GSO are supposed of glow worms that carry a luminescence quantity called luciferin along with them. The glow worms register the fitness of their current locations, evaluated using the objective function, into a luciferin value that they broadcast to their neighbours. GSO always understands its neighbours action and computes its movements by exploiting an adaptive neighbourhood, which is bounded above by its sensor range. Each slowworm in the algorithm has to be selected using a probabilistic mechanism, a neighbour that has a greater luciferin value than its own will be moves toward it. The glow worm's movement of each swarm is always depending on local information of its neighbours and selective neighbour interactions that enables the partition into disjoint subgroups that converge on multiple optima of a given multimodal function.

At the beginning, a light whose intensity is proportional to the associated luciferin and interact with other agents within a variable neighbourhood emitted by the glow worms. In each case the neighbourhood of the worm is defined as a local-decision domain that has a variable neighbourhood range v_d^ibounded by a radial sensor range r_s ($0 < v_d^i \leq r_s$). A glow worm $i$considers another glow worm $j$as its neighbour if j is within the neighbourhood range of $i$and the luciferin level of $j$is higher than that of i. From this decision domain the sub-swarms enable selective neighbour interactions and aids in formation of disjoint. There is a communication between neighbourhoods of worms depending on their brightness $i.e$each glow worm is attracted by the brighter glow of other glow worms in the neighbourhood. The individual agents in GSO has to get its neighbourhood information available to make decisions [59].

Generally, in the GSO Algorithm there are three main phases: luciferin update phase, movement phase, and decision range update. The luciferin update depends on the functional value of the glow worm at certain position. Even though, all glow worms begin movement with the same luciferin value at the initial iteration, these values are changing according to the function values at their current positions. This luciferin value has direct relation with the measured value of the sensed profile (temperature, radiation level) at that location. The luciferin is added to its previous level by each glow worm and at the same time, it is subtracted the previous luminescence value to simulate the decay in luminescence. The luciferin update rule is given by:

$$l_i(t+1) = (1-\rho)l_i(t) + \gamma J_i(t+1) \quad (1)$$

Where $l_i(t)$, represents the luciferin level associated with glow worm$i$ at time t, ρ is the luciferin decay constant $0 < \rho < 1$, c is the luciferin enhancement constant, and J_i represents the value of objective function at agent $i's$ location at time$t$.

In case of movement phase, each glow worm uses a probabilistic approach to decide a movement of a neighbour that has a luciferin value more than its own. As mentioned above, attraction of Glow worms is depending on neighbours that glow brighter. For each glow worm i, the probability of moving toward a neighbour worm j is given by:

$$P_{ij} = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)} \quad (2)$$

Where $j \in N_i(t), N_i(t) = \{ j: d_{i,j}(t) < v_d^i(t); l_i(t) < l_j(t)\}$is the set of neighbourhoods of glow worm i at time t. $d_{i,j}(t)$ represents the Euclidean distance between glow worms i and $j$at time t, and v_d^i represents the variable neighbourhood range associated with glow worms i at time t.

Neighbourhood range update rule: We associate with each agent$i$ a neighbourhood whose radial range v_d^i is dynamic in nature$0 < v_d^i \leq r_s$):r_s represents the radial range of the luciferin sensor. The fact that a fixed neighbourhood range is not used needs some justification. When the glow worms depend only on local information to decide their movements, it is expected that the number of peaks captured would be a function of the radial sensor range. In fact, if the sensor range of each agent covers the entire search space, the entire agents move to the global optimum and the local optima are ignored [58, 59].

3.4.9 Grey Wolf Optimization

Grey wolf optimization (GWO) is recently developed population based meta-heuristics algorithm that that takes its inspiration from the social hierarchy of hunting behaviour and leadership mechanism of grey wolves in nature proposed by Mirjalili et al. in 2014[60].

Mostly the living style of grey wolves is in a pack and one of the most important feature in their living style is very strict social hierarchy. The main leader of the pack which can be a male or female is called alphas and it is the most predominant wolf in the pack on making decisions about hunting, sleeping place, time to wake, and so on and their orders were followed by rest of the pack. The alpha wolf is one of the most important member in terms of managing the pack. The second level of the hierarchy is called beta wolves and they are subordinate wolves, which assist the alpha in making decisions and other pack activities. This category of wolf can be either male or female, and he/she is probably the best candidate the group to be the alpha in case one of the alpha wolves passes away or becomes very old. The beta wolf should respect the alphas decision, but commands the other lower-level wolves as well. It is an intermediate of the pack and plays the role of an advisor to the alpha and discipliner for the pack [60].

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

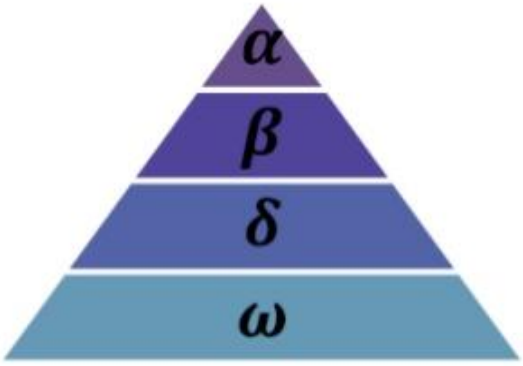

*Fig 2. Hierarchy of grey wolf [*61]

The lowest ranking grey wolves of the hierarchy are Omegas and they have to submit to all other dominant wolves. Delta wolves come in the hierarchy next to the alphas and betas but they lead the omega nearby. To mathematically model the social hierarchy of grey wolves inspired by nature, first the fitness solutions are determined from the problem and this best fitness solution is regarded as alpha (α), the second and third best solutions of the populations are considered as beta (β) and delta (δ), while the rests of the fitness solutions are regarded as omega (ω) wolves. Another very important thin in the grey wolves is that their hunting mechanism which includes tracking, chasing, encircling and harassing the prey until they stop moving [60,61,62]].

According to Muro*et al.* [62] the main categories of grey wolf hunting are as follows:

- Tracking, chasing, and approaching the prey.
- Pursuing, encircling, and harassing the prey until it stops moving.
- Attack towards the prey.

The Grey wolf optimization (GWO) was designed from natural activity of wolves and designing the above hunting process $i.e$Encircling Prey, harassing the prey and Attack towards the prey by Mathematical model it becomes powerful algorithms to solve real world problems.

a) Prey Encircling: The pack encircles a prey by repositioning individual agents according to the prey location, as follows:

$$\vec{X}(t+1) = \vec{X}_p(t) + \vec{A} . \vec{D} \quad (1)$$

where *t*is the iteration, $\vec{X}_p$is the prey position, $\vec{X}$is the grey wolf position, the (.) operator indicates vector entry wise multiplication, and $\vec{D}$is defined as follows:

$$\vec{D} = | \vec{C} . \vec{X}_p(t) - \vec{X}(t) | \quad (2)$$

where $\vec{A} and \vec{C}$are coefficient vectors calculated as follows:

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

$$\vec{D} = 2a \,.\vec{r}_1 - a \tag{3}$$

$$\vec{C} = 2\,\vec{r}_2 \tag{4}$$

where a is linearly diminished over the course of iterations controlling *exploration* and *exploitation*, and $\vec{r}_1$ and $\vec{r}_2$are random vectors in the range of [0, 1]. The value of a is the same for all wolves. These equations indicate that a wolf can update its position in the search space around the prey in any random location.

b) Hunting: It is performed by the whole pack based on the information coming from the *alpha*, *beta*, and *delta* wolves, which are expected to know the prey location, as given in the following:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{5}$$

Where $\vec{X}_1,\ \vec{X}_2 and \vec{X}_3$ are defined as follows

$$\begin{aligned} \vec{X}_1 &= |\,\vec{X}_\alpha - \vec{A}_1.\vec{D}_\alpha| \\ \vec{X}_2 &= |\,\vec{X}_\beta - \vec{A}_2.\vec{D}_\beta| \\ \vec{X}_3 &= |\,\vec{X}_\gamma - \vec{A}_3.\vec{D}_\gamma| \end{aligned} \tag{6}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\gamma$are the first three best solutions at a given iteration *t*, $\vec{A}_1$, $\vec{A}_2$, and $\vec{A}_3$are defined as for A above and $\vec{D}_\alpha, \vec{D}_\beta and \vec{D}_\gamma$ are defined using the following:

$$\begin{aligned} \vec{D}_\alpha &= |\vec{C}_1\,.\vec{X}_\alpha - \vec{X}| \\ \vec{D}_\beta &= |\vec{C}_2\,.\vec{X}_\beta - \vec{X}| \\ \vec{D}_\delta &= |\vec{C}_3\,.\vec{X}_\delta - \vec{X}| \end{aligned} \tag{7}$$

where $\vec{C}_1, \vec{C}_2 and \vec{C}_3$ are defined as for C above.

This is interpreted by the fact that *alpha*, *beta*, and *delta* wolves know the best position of the prey and all the other wolves adapt their positions based on the position of these wolves.

c) Attacking Stage: The agents approach the prey, which is achieved by decrementing the exploration rate *a*. Parameter *a* is linearly updated in each iteration to range from 2 to 0 as follows

$$a = 2 - t\frac{2}{T_{iter}} \tag{8}$$

where *t*is the iteration number and T_{iter} is the total number of iterations allowed for the optimization

d) *Search for Prey:* Wolves diverge from each other to search for prey. This behaviour is modelled by setting large values for parameter *a* to allow for exploration of the search space. Hence, the wolves diverge from each other to better explore the search space and then converge again to attack when they find a better prey. Any wolf can find a better prey (optimum). If they get closer to the prey, they will become the new *alphas* and the other wolves will be split into *beta*, *delta*, and *omega* according to their distance from the prey.

To see how GWO is theoretically able to solve optimization problems, some points may be noted: -

- The proposed social hierarchy assists GWO to save the best solutions obtained so far over the course of iteration.
- The proposed encircling mechanism defines a circle-shaped neighbourhood around the solutions which can be extended to higher dimensions as a hyper-sphere.
- The random parameters *A* and *C* assist candidate solutions to have hyper-spheres with different random radii.
- The proposed hunting method allows candidate solutions to locate the probable position of the prey.
- Exploration and exploitation are guaranteed by the adaptive values of *a* and *A*.
- The adaptive values of parameters *a* and *A* allow GWO to smoothly transition between exploration and exploitation.
- With decreasing *A*, half of the iterations are devoted to exploration ($|A| \geq 1$) and the other half are dedicated to exploitation ($|A| < 1$).
- The GWO has only two main parameters to be adjusted (*a* and *C*).

3.5 Water Wave Optimization

Water wave optimization (WWO) is a fresh newly introduced evolutionary algorithm borrowing ideas from shallow water wave models for global optimization problems. Zheng proposed a novel optimization algorithm called water wave optimization (WWO) [50], which derives effective mechanisms from water wave phenomena such as propagation, refraction, and breaking for solving high-dimensional global optimization problems. Numerical tests and real-world problems applications have shown that WWO is very inexpensive to other state-of-the-art heuristic algorithms. WWO is originally proposed for global optimization in continuous search spaces. However, as other global optimization methods such as PSO and DE that have been adapted to solve many combinatorial optimization problems, we believe that WWO also has a great potential in combinatorial optimization.

WWO [5] is a global optimization algorithm, which uses numerical techniques to study the evolution of wave heights, periods, and propagation directions under various conditions such as wind forcing, nonlinear wave interactions, and frictional dissipation. In WWO, the solution space is analogous to the seabed area of natural process and the fitness of a point in the space is

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: http://www.ijesm.co.in, Email: ijesmj@gmail.com Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

measured inversely proportional to its seabed depth:$i.e$ the shorter the distance to the still water level, the higher the fitness value it gives. It should be well-known that by analogy the 3-D space of the seabed is generalized to a high-dimensional space. As most evolutionary algorithms, WWO maintains a population of solutions, each of which is analogous to a wave with a height h in the integer domain and a wavelength λ in the real domain (initialized as a constant h_{max} and 0.5 respectively). A good wave (high quality solution) located in shallow water has a large height (high energy) and a short wavelength, while a bad wave (poor quality solution) located in shallow water has a small height (low energy) and a long wavelength, as illustrated in Fig. 1. The search process of the WWO algorithm in the solution space is modelled as the propagation, refraction, and breaking of the waves [51].

Propagation

In propagation, each wave x propagates exactly once at each generation, which creates a new wave x' by shifting each dimension d of the original wave x as follows:

$$x^{'}(d) = x(d) + rand(-1,1)\,.\lambda.L \qquad (1)$$

Where: - rand1,1) is a random number uniformly distributed in the range [-1,1], and L(d) is the length of the d^{th} dimension of the search space $(1 \leq d \leq n)$.

If the new position is outside the feasible range, it will be reset to a random position in the range. After propagation, if the fitness of the offspring wave $x'$is higher than the original x, then x is replaced by x' whose height is reinitialized to h_{max}. Otherwise, the height of the original x is decreased by one. In this way, high fitness waves tend to perform more local search in small areas and low fitness waves tend to perform more global search in wide areas. After each generation, each wave$x$ updates its wavelength as follows:

$$\lambda = \lambda\,.\,\alpha^{-(f(x)-f_{min}+\varepsilon)/(f_{max}-f_{min}+\varepsilon)} \qquad (2)$$

Where $f_{max}\,and f_{min}$ are the maximum and minimum fitness values in the current population respectively, α is a parameter called the wavelength reduction coefficient, and ε is a small positive number to avoid division-by-zero.

Refraction

WWO uses the wave height property$h$ to control the life span of the waves: If a wave propagates from deep water (low fitness location) to shallow water (high fitness location), its height increases; otherwise its height decreases, which mimics energy dissipation. When the height of a wave$x$ decreases to zero, WWO uses the refraction operator to replace x with a new generated wave x'. Each dimension of x' is set as a random number cantered halfway between the original position and the corresponding position of the best-known solution x^*:

$$x^{'}(d) = norm\,(\frac{x^*(d)+x(d)}{2}\,,\,\frac{|x^*(d)-x(d)|}{2}) \qquad (3)$$

International Journal of Engineering, Science and Mathematics
Vol. 6 Issue 7, November 2017,
ISSN: 2320-0294 Impact Factor: 6.238
Journal Homepage: **http://www.ijesm.co.in, Email: ijesmj@gmail.com** Double-Blind Peer
Reviewed Refereed Open Access International Journal - Included in the International Serial Directories Indexed & Listed at: Ulrich's Periodicals Directory ©, U.S.A., Open J-Gage as well as in Cabell's Directories of Publishing Opportunities, U.S.A

Where norm(μ, σ) is a Gaussian random with mean μ and standard deviation σ.

The purpose of the operator is to remove those waves that fail to find better solutions for a given period so as to avoid search stagnation, and generates new waves to improve the diversity. After each refraction operation, the height of new x' is also set to h_{max} , and its λ is updated as:

$$\lambda' = \lambda \frac{f(x)}{f(x')} \quad (4)$$

Breaking

The third operator in WWO is breaking, which simulates the breaking of a steepest wave into a train of solitary waves so as to enhance local search around a promising point. Whenever a new wave x is found to be better than the best-known solution $x*$ (i.e., $x$becomes a new current best), WWO applies the breaking operation to conduct local search around x several times. In details, the breaking operator first randomly chooses k dimensions of x, and then generates a solitary wave x' at each chosen dimension d as follows:

$$x'(d) = x(d) + norm(0,1) \,.\beta.L(d) \quad (5)$$

Where β is a parameter called the breaking coefficient, k is a random number between 1 and a predefined numberk_{max} . Empirically, it is suggested to set k_{max} as min (12, $\frac{n}{2}$), where n is the dimension of the problem. If none of the solitary waves is better than x, x is remained; else it is replaced by the best one among the solitary waves.

4. Comparative Analysis

Most of the Optimization problems in the real-world are often very challenging to solve and many applications have to deal with NP-hard problems in the past decades. To solve such problems, there are a number of technique developed by different scholars. These Optimization techniques are broadly classified as conventional (deterministic) and non-conventional (stochastics) which are applied to solve the given problem based on the problems on hand. Numerous conventional optimization techniques are pre-designed algorithmic formulas and have been designed to solve a wide range of optimization problems, such as Linear Programming, Nonlinear Programming, Dynamic Programming or Combinatorial Optimization. But, many of these existing exact or conventional optimization techniques applied to real-world problems suffer a lot of problems such as: difficulties in passing over local optimal solutions; risk of divergence; difficulties in handling constraints or numerical difficulties related to computing first or second order derivatives. These are due to increase the number of the variables in the problem, non-linearity of the problem, non-differentiability of the problems or totally stochastic nature of the problems. All of such techniques are derivative dependant.

To overcome these problems, the non-conventional (stochastics) techniques were proposed in the early 70‟. Unlike exact methods, the stochastic methods are simple to apply and more power full to locate the global optimal of the given problem by being often based on criteria of empirical nature. These techniques are responsible for the absence of any guarantee for successfully identifying the optimal

solution [41]. Even though, it is a challenging and impossible task to classify these algorithms in detail systematically in to distinct groups, we try to classify from the view of source of inspirations to encourage researcher to give more attention on the distinct classifications.

The sources nature inspired algorithms are from natural evolution, swarm movement of animals or insects and also from natural ecology. All these algorithms are effectively employed when the conventional methods are fails to solve a problem at hand due to a number of reasons.

Evolutionary algorithms are all population-based stochastic search algorithms performing with best-to-survive criteria. They start by creating an initial population of feasible solutions and continuous the process of iteration improvement from generation to generation towards a best solution. Relatively computationally cost and if not properly handled there is no guarantee for global solutions. Where as Swarm Intelligence algorithms are inspired from collective behaviour of agents who follow some simple rules which can accomplish very trajectory work in group. The later one is computationally simple and take advantage over EA in many circumstances.

Table 1. Comparative analysis between conventional and non-conventional techniques

Techniques	Problems for Computational formula or algorithms available		Problems for computational formula fails or difficult to model	
	Exact solution	Time needed	Optimal solution	Time needed
Conventional	VG	VG	P	P
Non-conventional	P	P	VG	VG

The values for the given expressions are, poor (P), Fair (F), Good (G) and Very Good (VG).

Table 2. Comparative Analysis of nature inspired Algorithms

Algorithm	Mimics biological evolution	Based on collective behavior of organisms	Efficient to solve hard Optimization problems
GA	√	x	√
GP	√	x	√
EA	√	x	√
DE	√	x	√
PSO	x	√	√

Firefly	x	√	√
Water Wave	x	√	√
BAT	x	√	√
CODA	x	√	√
ACO	x	√	√
Cuckoo Search	x	√	√
Grey Wolf	x	√	√

The value (√) given for correct/ high and (x) for incorrect/ low

5. Conclusion

Computational Intelligence is a set of nature-inspired computational techniques and methodologies and approaches to address complex real-world problems to which mathematical or traditional modelling are failed due to: the processes might be too complex for mathematical reasoning, it might contain some uncertainties during the process, or the process might simply be stochastic in nature. In addition to these, many real-world problems around us cannot be translated into binary language for computers to process it. For such difficulty, Computational Intelligence was introduced and handling the problems in multi directions. As many literatures shows that researchers have focused their attention on innovative use of CI techniques in multi real life problems.

CI is a combination of five main complementary techniques. The fuzzy logic which enables the computer to understand natural language , artificial neural networks which permits the system to learn experiential data by operating like the biological one, evolutionary computing, which is based on the process Biological Organism and Collective behaviour of organisms or animals. learning theory, and probabilistic methods which helps dealing with uncertainty imprecision.

In this survey paper, Techniques of Computational intelligence and their applications are explained by focusing on Evolutionary Computing which contains population based bio inspired algorithms and swarm based algorithms. The application of these techniques is depending the nature the problems. Researchers identify the proper technique of their problem by considering variables and parameters of the problems including the purpose and time needed to compute the

problem. Many literatures have shown that, Swarm intelligence algorithms are newly emerging techniques and relatively simple for computations. But, the hybrid of these techniques didn't focus yet and needs great attention of the researchers to get more power full technique of the field for future computing. We hope this document give high light of nature inspired computational techniques with their advantages, limitations and application areas are complied with their full references to show the interested reader to get more detailed about the concepts raised here.

Acknowledgements

Authors are grateful to Punjabi University Patiala for providing adequate library and internet facility.

References

[1] Duch Włodzisław.What is computational intelligence and what could it become. *Computational Intelligence, Methods and Applications Lecture Notes NTU, Singapore,* (2003).

[2] Siddique Nazmul and Hojjat Adeli. *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*. John Wiley & Sons (2013).

[3] Bezdek James C. On the relationship between neural networks, pattern recognition and intelligence, *International journal of approximate reasoning*, Vol. 6, no. 2, pp. 85-107 (1992).

[4] Siddique Nazmuland Hojjat Adeli. *Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing*. John Wiley & Sons (2013).

[5] Zadeh, L.A. Fuzzy Sets. Information and Control Vol. 61, No. 9, pp. 338 -358 (1965).

[6] Grossberg Stephen. Nonlinear neural networks: Principles, mechanisms, and architectures, *Neural networks*, Vol.1, no. 1, pp. 17-61 (1988).

[7] Rumelhart David E. Bernard Widrow and Michael A. Lehr., The basic ideas in neural networks, *Communications of the ACM, Vol.* 37, no. 3, pp. 87-93 (1994).

[8] Somers Mark John and Jose C. Casal. Using artificial neural networks to model nonlinearity: The case of the job satisfaction job performance relationship, *Organizational Research Methods, Vol.* 12, no. 3, pp. 403-417 (2009).

[9] Ormrod Jeanne Ellis.*Educational psychology: Principles and applications*, Merrill, (1995)

[10] Illeris Knud. A model for learning in working life, *Journal of workplace learning*, Vol.16, no. 8, pp. 431-441 (2004).

[11] Erdos Paul and Joel Spencer. Probabilistic methods in combinatorics., *AMC* Vol.10, no. 12 (1974).

[12] Franz T., R. Kothary, M. A. H. Surani, Z. Halata and M. Grim. The Splotch mutation interferes with muscle development in the limbs., *Anatomy and embryology*, Vol.187, no. 2, pp. 153-160 (1993).

[13] Pallit A., and D. Popovic. Computational Intelligence in Time Series Forecasting, (2005).

[14] Dixit Manish, Nikita Upadhyay and Sanjay Silakari. An exhaustive survey on nature inspired optimization algorithms, *International Journal of Software Engineering and Its Applications*, Vol.9, no. 4, pp. 91-104 (2015).

[15] BinithaS., and S. Siva Sathya.A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering*, Vol.2, no. 2, pp. 137-151 (2012).

[16] T. Bäck. Evolutionary Algorithms in Theory and Practice,Oxford, New York, 1996.

[17] HollandJohn H. Genetic algorithms and the optimal allocation of trials, *SIAM Journal on Computing*, Vol.2, no. 2, pp. 88-105 (1973).

[18] HollandJohn Henry. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*., MIT press, (1992).

[19] Yang X.S. Nature-inspired metaheuristic algorithms., *Luniver press,* 2010..

[20] X. Yang. Nature-Inspired Metaheuristic Algorithms, *Luniver Press*, vol. 41, (2008).

[21] Bramlette, Mark F. and Eugene E. Bouchard. Genetic algorithms in parametric design of aircraft., *Handbook of genetic algorithms, pp.* 109-123 (1991).

[22] Harp Steven Alex and Tariq Samad. Genetic synthesis of neural network architecture., *Handbook of genetic algorithms, pp.* 202-221 (1991).

[23] Stadlthanner K., Dominik L., Fabian J. T., Elmar Wolfgang Lang, Ana Maria Tomé, Petia Georgieva, and Carlos García Puntonet., Sparse nonnegative matrix factorization with genetic algorithms for microarray analysis, In *Neural Networks., International Joint Conference on*, pp. 294-299. IEEE, (2007).

[24] Koza, John R., *Genetic programming: on the programming of computers by means of natural selection*. Vol. 1. MIT press, (1992).

[25] Worzel William P., Jianjun Y., Arpit A. Almal, and Arul M. C., Applications of genetic programming in cancer research., *The international journal of biochemistry & cell biology*, Vol.41, no. 2, pp.405-413. (2009).

[26] Espejo Pedro G., Sebastián Ventura, and Francisco Herrera., A survey on the application of genetic programming to classification., *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), Vol.* 40, no. 2, pp. 121-144 (2010).

[27] Beyer, H. G., & Schwefel, H. P., Natural Computing 1, in *Evolution strategies*, Springer, pp. 3-52 (2002).

[28] Storn Rainer., Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, *Technical report, International Computer Science Institute*, Vol.11 (1995).

[29] Storn Rainer and Kenneth Price., Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces., *Journal of global optimization. Vol.* 11, no. 4, Pp. 341-359 (1997).

[30] Eberhart, R. C. and J. Kennedy., Particle swarm optimization, proceeding of IEEE International Conference on Neural Network, *Perth, Australia, pp.* 1942-1948 (1995).

[31] Eberhart R. C., and J. Kennedy., Particle swarm optimization, proceeding of IEEE International Conference on Neural Network, *Perth, Australia, PP.*1942-1948 (1995).

[32] Gharehchopogh F. S., Zahra A. Dizaji, and Zahra Aghighi., Evaluation of particle swarm optimization algorithm in prediction of the car accidents on the roads: A case study., *International Journal on Computational Sciences & Applications (IJCSA), Vol.* 3, no. 4, pp. 1-12 (2013).

[33] Dheeba, J. and Tamil Selvi., Bio inspired swarm algorithm for tumour detection in digital mammogram., *Swarm, Evolutionary, and Memetic Computing, pp.* 404-415 (2010).

[34] Jona, J., and N. Nagaveni., A hybrid swarm optimization approach for feature set reduction in digital mammograms., *WSEAS Transactions on Information Science and Applications, Vol.* 9, pp. 340-349 (2012).

[35] Navin, A., and M. Mirnia., A new algorithm to classify face emotions through eye and lip features by using particle swarm optimization, In *2012 4th International Conference on Computer Modeling and Simulation (ICCMS 2012) IPCSIT*, vol. 22, pp. 268-274 (2012).

[36] Dorigo, Marco, Vittorio Maniezzo, and Alberto Colorni., Ant system: optimization by a colony of cooperating agents., *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), Vol.* 26, no. 1, pp. 29-41. (1996).

[37] Dorigo Marco and Luca Maria Gambardella., Ant colony system: a cooperative learning approach to the traveling salesman problem., *IEEE Transactions on evolutionary computation, Vol*, 1, no. 1, pp. 53-66 (1997).

[38] S. N. Tazi, Mukesh Gupta and Akansha Jain, A Survey On Application Of Nature Inspired Algorithms, *International Journal of Computer Science Engineering*, vol. 4, no. 4, pp. 33-40, (2014).

[39] X.S. Yang, A new metaheuristic bat-inspired algorithm. In: González, J.R., Pelta, D.A., Cruz, C., Terrazas, G., Krasnogor, N. (eds.) Nature inspired cooperative strategies for optimization (NICSO 2010),*Studies in Computational Intelligence*, vol. 284, p. pp. 65–74 (2010).

[40] Yang Xin-Sh, Bat algorithm for multi-objective optimisation., *International Journal of Bio-Inspired Computation, Vol.* 3, no. 5, pp. 267-274 (2011).

[41] Manshahia, M.S., Dave, M. and Singh, S.B., Bio Inspired Congestion Control Mechanism for Wireless Sensor Networks, I*n: Proceedings. of 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Madurai, India , December, 2015.

[42] Manshahia, M.S., Dave, M. and Singh, S.B, Improved Bat Algorithm Based Energy Efficient Congestion Control Scheme for Wireless Sensor Networks, *Wireless Sensor Network*, Vol. 8, pp. 229-241(2016).

[43]Yang, Xin-She, *Nature-inspired metaheuristic algorithms*. Luniver press, (2010).

[44] Yang Xin-She, Seyyed S. S. and Amir Hossein G., Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect, *Applied soft computing*, Vol.12, no. 3, pp. 1180-1186 (2012).

[45] Manshahia, M.S., A Firefly Based Energy Efficient Routing in Wireless Sensor Networks, *African Journal of Computing & ICT*, Vol. 8, No. 4, p. 27-32 (2015)

[46] Manshahia, M.S., Dave, M. and Singh, S.B., Firefly algorithm based clustering technique for Wireless Sensor Networks, I*n: Proceedings of International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET),* 23-25 March 2016.

[47] Manshahia, M.S., Dave, M. and Singh, S.B., Congestion Control in Wireless Sensor Networks Based on Bioluminescent Firefly Behavior, *Wireless Sensor Network*, Vol. 7, p. 149-156 (2015).

[48] Banati Hema and Monika Bajaj, Firefly based feature selection approach. *International Journal of Computer Science Issues, Vol.* 8, no. 4, pp. 473-480 (2011).

[49]Honarpisheh Zahra and Karim Faez., An efficient dorsal hand vein recognition based on firefly algorithm., *International Journal of Electrical and Computer Engineering (IJECE), Vol.* 3, no. 1, pp. 30-41 (2013).

[50] Das, Hillol, Ashim Saha, and Suman Deb., An expert system to distinguish a defective eye from a normal eye, In *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*, pp. 155-158. IEEE, (2014).

[51] Karaboga Dervis, and Bahriye Basturk., A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of global optimization, Vol.* 39, no. 3, pp. 459-471 (2007).

[52] Bolaji Asaju La'aro, Ahamad Tajudin Khader, Mohammed Azmi Al-Betar, and Mohammed A. Awadallah., Artificial bee colony algorithm, its variants and applications: A survey, *Journal of Theoretical & Applied Information Technology*, Vol.47, no. 2 (2013).

[53] Hancer Emrah, Celal Ozturk, and Dervis Karaboga., Artificial bee colony based image clustering method., In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pp. 1-5. IEEE, (2012).

[54]Tuba Milan, Milos Subotic, and Nadezda Stanarevic., Modified cuckoo search algorithm for unconstrained optimization problems, In *Proceedings of the 5th European conference on European computing conference*, World Scientific and Engineering Academy and Society (WSEAS), pp. 263-268 (2011).

[55]Mohamad Azizah Binti, Azlan Mohd Zain and Nor Erne Nazira Bazin., Cuckoo search algorithm for optimization problems a literature review and its applications, *Applied Artificial Intelligence, Vol.* 28, no. 5, pp. 419-448 (2014).

[56]Gandomi Amir Hossein, Xin-She Yang and Amir Hossein Alavi., Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems." *Engineering with computers, Vol.* 29, no. 1, pp. 17-35 (2013).

[57]Li, Xiao-lei., An optimizing method based on autonomous animats: fish-swarm algorithm." *Systems Engineering-Theory & Practice, Vol.* 22, no. 11, pp. 32-38. (2002).

[58]Wu Bin, Cunhua Qian, Weihong Ni, and Shuhai Fan., The improvement of glow worm swarm optimization for continuous optimization problems., *Expert systems with applications, Vol* 39, no. 7, pp. 6335-6342 (2012).

[59] Krishnanand, K. N., and Debasish Ghose., Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions., *Swarm intelligence, Vol.* 3, no. 2, pp. 87-124 (2009).

[60] Mirjalili Seyedali, Seyed Mohammad Mirjalili, and Andrew Lewis., Grey wolf optimizer, *Advances in Engineering Software, Vol.* 69, pp. 46-61 (2014).

[61] Sharma Sharmistha, Subhadeep Bhattacharjee and Aniruddha Bhattacharya., Grey wolf optimisation for optimal sizing of battery energy storage device to minimise operation

cost of microgrid, *IET Generation, Transmission & Distribution, Vol.* 10, no. 3, pp. 625-637(2016).

[62] Muro C., R. Escobedo L. Spector and R. P. Coppinger., Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. *Behavioural processes, Vol.* 88, no. 3, pp. 192-197 (2011).

[63]Zheng Yu-Jun., Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research*, Vol. 55, pp. 1-11 (2015).

[64] Wu Xiao-Bei, Jie Liao, and Zhi-Cheng Wang., Water wave optimization for the traveling salesman problem., In *International Conference on Intelligent Computing*, Springer, Cham, pp. 137-146 (2015).

[65] Manshahia, M.S., Water Wave Optimization Algorithm based Congestion Control and Quality of Service Improvement in Wireless Sensor Networks, *Transactions on Networks and Communications*, Vol. 5, No. 4, p .31-39 (2017)

[66] Manshahia, M.S., Wireless Sensor Networks: A Survey, *International Journal of Scientific & Engineering Research*, Vol. 7, No. 4, p. 710-716 (2016).

[67]Mukhdeep Singh Manshahia, Mayank Dave, S.B. Singh, Computational Intelligence for Congestion Control and Quality of Service Improvement in Wireless Sensor Networks, *Transactions on Machine Learning and Artificial Intelligence*, Vol. 5, No. 6, pp: 21-35 (2017).