A Survey of Nature Inspired Clever Algorithms

Manas Kumar Yogi[1] Darapu Uma[2]
[1,2]Assistant Professor
[1,2]Department of Computer Science and Engineering
[1,2]Pragati Engineering College, Surampalem,Kakinada,A.P.

Abstract— This paper presents unconventional algorithms deployed for search, optimization called as Clever algorithms. They take input from environment and based on multiheuristics, formulate candidate solutions to solve the problem in hand. Stochastic Algorithms that concentrates on the introduction of randomness into heuristic methods. Evolutionary Algorithms inspired by evolution by means of natural selection. Physical Algorithms inspired by physical and social systems. Swarm Algorithms that relates to methods that exploit the properties of collective intelligence. Immune Algorithms inspired by the adaptive immune system of vertebrates. Our paper sincerely attempts to provide a glimpse of Clever algorithms to understand the metaphor and underlying mechanism of such bio-inspired algorithms.

Key words: Clever, Stochastic, immune, evolutionary, swarm, intelligence

I. INTRODUCTION

Algorithm refers to the procedural logic to achieve a certain solution for the problem in hand. Clever algorithm is a term coined in order to address ambiguously described algorithms or inconsistent algorithms. Inconsistent in the sense its extensibility is limited and transferring it into different environment fails the logic present in that algorithm. For users of an algorithm, no gaps must be present in the description else such algorithm cannot be put to use. Nature inspired algorithms mimic the behavior of animate objects whether its human behavior, animal behavior, behavior of plants etc. Clever algorithms are used in fields of computational intelligence and other relevant fields. The aim of this field is to construct mathematical as well as engineering tools to produce solutions to computation problems. The field includes using procedures for finding solutions abstracted from the natural world for addressing computationally phrased problems. For implementing clever algorithms there are popular programming paradigms like procedural, object oriented, flow programming. In this paper we try to present a clear vision about the fundamental concepts inherent in a clever algorithms. We have surveyed the class of clever algorithms including evolutionary algorithms, swarm algorithms, immune algorithms, physical algorithms, stochastic algorithms.

II. TAXONOMY

In this section we present the taxonomy of clever algorithms. Evolutionary algorithms as proposed by Darwin involve principles of biological process evolution like natural selection and propagation of genetic material in the process to account for adaptive fit in an environment. These algorithms are iterative in nature and include mechanisms of recombination, mutation of processes to obtain candidate solutions in the environment.

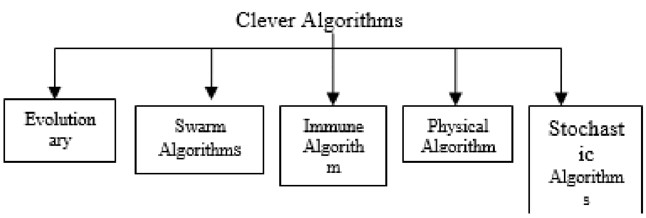

Fig. 1: Classification of Clever Algorithms

Swarm algorithms are derived from collective intelligence systems where intelligence is emerged from numerous homogeneous agents in the environment. For instance consider schools of fish, flocks of birds, and colonies of ants. Here intelligence is decentralized, self-organizing and distributed throughout an environment. In nature such systems are generally utilized to solve problems such as effective collection for food, prey evasion, colony re-location etc. The information is typically stored throughout the participating homogeneous gents, or is stored or communicated in the environment itself such as through the use of pheromones in ants, dancing in bees, and proximity in fish and birds.

Immune Algorithms belong to the Artificial Immune Systems field of study related with computational methods motivated by the process and techniques of the biological immune system. A simplified description of the immune system is an organ system intended to protect the host organism from the threats posed to it from pathogens and toxic substances. Pathogens includes a range of microorganisms such as bacteria, viruses, parasites and pollen. The general viewpoint about the role of the immune system is divided into two primary tasks: the detection and elimination of pathogen. This behavior is referred to as the differentiation of self (molecules and cells that belong to the host organisms) from potentially harmful non-self. More recent implementations for the role of the system include a maintenance system and a cognitive system .The architecture of the immune system is such that a series of defensive layers protect the host. Once a pathogen makes it inside the host, it must contend with the innate and acquired immune system. These interrelated immunological sub-systems are comprised of many types of cells and molecules produced by specialized organs and processes to solve the self-nonself problem at the lowest level taking help of chemical bonding, where the surfaces of cells and molecules interact with the surfaces of pathogen.

Physical algorithms are those algorithms inspired by a physical process. The described physical algorithm generally belong to the fields of Met heuristics and Computational Intelligence, although do not fit neatly into the existing categories of the biological inspired techniques (such as Swarm, Immune, Neural, and Evolution). In this regard, they could be referred to as nature inspired algorithms. The inspiring physical systems range from metallurgy, music, the interplay between culture and evolution, and complex dynamic systems such as avalanches. They are generally stochastic optimization algorithms with a mixtures of local (neighborhood-based) and global search techniques.

Stochastic algorithms use randomness. They use all combinations but not in order but instead they use random ones from the whole range of possibilities hoping to hit the solution sooner. Implementation is fast easy and single iteration is also fast (constant time). These described algorithms are predominately global optimization algorithms and metaheuristics that manage the application of an embedded neighbourhood exploring (local) search procedure. As such, with the exception of 'Stochastic Hill Climbing' and 'Random Search' the algorithms may be considered extensions of the multi-start search. This set of algorithms provide various different strategies by which 'better' and varied starting points can be generated and issued to a neighbourhood searching technique for refinement, a process that is repeated with potentially improving or unexplored areas to search .

A. Evolutionary Algorithms:

In this section we concentrate on 3 types of algorithm

1) .Genetic Algorithm:

The Genetic Algorithm is inspired by population genetics (including heredity and gene frequencies), and evolution at the population level, as well as the Mendelian understanding of the structure (such as chromosomes, genes, alleles) and mechanisms (such as recombination and mutation). This is the so-called new or modern synthesis of evolutionary biology.

Mechanism:

The aim of the Genetic Algorithm is to maximize the payoff of candidate solutions in the population against a cost function from the problem domain. The mechanism for the Genetic Algorithm is to repeatedly employ surrogates for the recombination and mutation genetic mechanisms on the population of candidate solutions, where the cost function (also known as aim or fitness function) applied to a decoded representation of a candidate governs the probabilistic contributions a given candidate solution can make to the subsequent generation of candidate solutions.

2) Non-Dominated Sorting Genetic Algorithm:

The aim of the NSGA algorithm is to improve the adaptive fit of a population of candidate solutions to a Pareto front constrained by a set of aim functions. The algorithm uses an evolutionary process with surrogates for evolutionary operators including selection, genetic crossover, and genetic mutation. The population is sorted into a hierarchy of subpopulations based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions.

3) Strength Pareto Evolutionary Algorithm:

The aim of the algorithm is to locate and maintain a front of non-dominated solutions, ideally a set of Pareto optimal solutions. This is achieved by using an evolutionary process (with surrogate procedures for genetic recombination and mutation) to explore the search space, and a selection process that uses a combination of the degree to which a candidate solution is dominated (strength) and an estimation of density of the Pareto front as an assigned fitness. An archive of the non-dominated set is maintained separate from the population of candidate solutions used in the evolutionary process, providing a form of elitism.

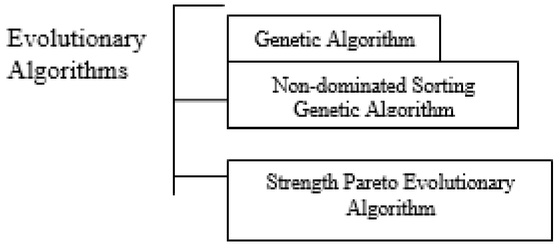

Fig. 2: Classification of Evolutionary Algorithms

B. Swarm Algorithms

The most famous and popular swarm algorithms are discussed below.

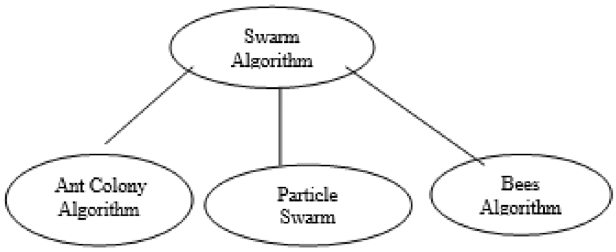

Fig. 3: Classification of Swarm Algorithms

1) Ant Colony Algorithm:

The Ant Colony System algorithm is represented by the foraging behavior of ants, specifically the pheromone communication between ants regarding a good path between the colony and a food source in an environment. This mechanism is called stigmergy .Ants initially wander randomly around their environment. Once food is located an ant will begin laying down pheromone in the environment. Numerous trips between the food and the colony are performed and if the same route is followed that leads to food then additional pheromone is laid down. Pheromone decays in the environment, so that older paths are less likely to be followed. Other ants may discover the same path to the food and in turn may follow it and also lay down pheromone. A positive feedback process routes more and more ants to productive paths that are in turn further refined through use. The aim of the mechanism is to exploit historic and heuristic information to construct candidate solutions and fold the information learned from constructing solutions into the history. Solutions are constructed one discrete piece at a time in a probabilistic step-wise manner. The probability of selecting a component is determined by the heuristic contribution of the component to the overall cost of the solution and the quality of solutions from which the component has historically known to have been included. History is updated proportional to the quality of the best known solution and is decreased proportional to the usage if discrete solution components.

2) Particle Swarm Optimization:

Particle Swarm Optimization is inspired by the social foraging behavior of some animals such as flocking behavior of birds and the schooling behavior of fish. Particles in the swarm fly through an environment following the fitter members of the swarm and generally biasing their movement toward historically good areas of their environment. The goal of the algorithm is to have all the particles locate the optima in a multi-dimensional hyper-volume. This is achieved by assigning initially random positions to all particles in the space and small initial random velocities. The algorithm is executed like a simulation, advancing the position of each particle in turn based on its velocity, the best known global position in the problem space and the best position known to a particle. The aim function is sampled after each position update. Over time, through a combination of exploration and exploitation of known good positions in the search space, the particles cluster or converge together around an optima, or several optima.

3) Bees Algorithm:

The Bees Algorithm is inspired by the foraging behavior of honey bees. Honey bees collect nectar from vast areas around their hive (more than 5 kilometers). Bee Colonies are observed to send bees to collect nectar from flower patches relative to the amount of food available at each patch. Bees communicate with each other at the hive via a waggle dance that informs other bees in the hive as to the direction, distance, and quality rating of food sources. Honey bees collects nectar from flower patches as a food source for the hive. The hive sends out scout's that locate patches of flowers, who then return to the hive and inform other bees about the fitness and location of a food source via a waggle dance. The scout returns to the flower patch with follower bees. A small number of scouts continue to search for new patches, while bees returning from flower patches continue to communicate the quality of the patch. The information processing aim of the algorithm is to locate and explore good sites within a problem search space. Scouts are sent out to randomly sample the problem space and locate good sites. The good sites are exploited via the application of a local search, where a small number of good sites are explored more than the others. Good sites are continually exploited, although many scouts are sent out each iteration always in search of additional good sites.

C. Immune Algorithms:

Here we present 4 algorithms which mimic the behavior of Artificial Immune Systems. Their division is as shown in figure below.

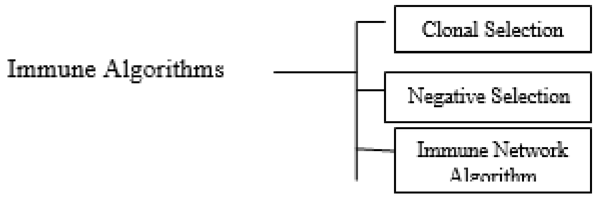

Fig. 4: Classification of Immune Algorithms

1) Clonal Selection Algorithm:

The theory suggests that starting with an initial repertoire of general immune cells, the system is able to change itself (the compositions and densities of cells and their receptors) in response to experience with the environment. Through a blind process of selection and accumulated variation on the large scale of many billions of cells, the acquired immune system is capable of acquiring the necessary information to protect the host organism from the specific pathogenic dangers of the environment. It also suggests that the system must anticipate (guess) at the pathogen to which it will be exposed and requires exposure to pathogen that may harm the host before it can acquire the necessary information to provide a defense. The information processing principles of the clonal selection theory describe a general learning mechanism. This mechanism involves a population of adaptive information units (each representing a problem-solution or component) subjected to a competitive processes for selection, which together with the resultant duplication and variation ultimately improves the adaptive fit of the information units to their environment. The general CLONALG model involves the selection of antibodies (candidate solutions) based on affinity either by matching against an antigen pattern or via evaluation of a pattern by a cost function. Selected antibodies are subjected to cloning proportional to affinity, and the hypermutation of clones inversely-proportional to clone affinity. The resultant clonal-set competes with the existent antibody population for membership in the next generation. In addition, low-affinity population members are replaced by randomly generated antibodies. The pattern recognition variation of the algorithm includes the maintenance of a memory solution set which in its entirety represents a solution to the problem. A binary-encoding scheme is employed for the binary-pattern recognition and continuous function optimization examples, and an integer permutation scheme is employed for the Travelling Salesman Problem (TSP).

2) Negative Selection Algorithm:

The Negative Selection algorithm is inspired by the self-nonself discrimination behavior observed in the mammalian acquired immune system. The clonal selection theory of acquired immunity accounts for the adaptive behavior of the immune system including the ongoing selection and proliferation of cells that select-for potentially harmful (and typically foreign) material in the body. An interesting aspect of this process is that it is responsible for managing a population of immune cells that do not select-for the tissues of the body, specifically it does not create self-reactive immune cells known as auto-immunity. This problem is known as 'self-nonself discrimination' and it involves the preparation and ongoing maintenance of a repertoire of immune cells such that none are auto-immune. This is achieved by a negative selection process that selects-for and removes those cells that are self-reactive during cell creation and cell proliferation. This process has been observed in the preparation of T-lymphocytes, naive versions of which are matured using both a positive and negative selection process in the thymus. The self-nonself discrimination principle suggests that the anticipatory guesses made in clonal selection are filtered by regions of infeasibility (protein conformations that bind to self-tissues). Further, the self-nonself immunological paradigm proposes the modeling of the unknown domain (encountered pathogen) by modeling the complement of what is known. This is unintuitive as the natural inclination is to categorize unknown information by what is different from that which is known, rather than guessing at the unknown information and filtering those guesses by what is known. The information processing principles of the self-nonself discrimination process via negative selection are that of an anomaly and change detection systems that model the anticipation of variation from what is known. The principle is achieved by building a model of changes, anomalies, or unknown (non-normal or non-self) data by generating patterns that do not match an existing corpus of available (self or normal) patterns. The prepared non-normal model is then used to either monitor the existing normal data or streams of new data by seeking matches to the non-normal patterns.

3) Immune Network Algorithm:

The Artificial Immune Network algorithm is inspired by the Immune Network theory of the acquired immune system. The clonal selection theory of acquired immunity accounts for the adaptive behavior of the immune system including the ongoing selection and proliferation of cells that select-for potentially harmful (and typically foreign) material in the body. A concern of the clonal selection theory is that it presumes that the repertoire of reactive cells remains idle when there are no pathogen to which to respond. Jerne proposed an Immune Network Theory (Idiotypic Networks) where immune cells are not at rest in the absence of pathogen, instead antibody and immune cells recognize and respond to each other. The Immune Network theory proposes that antibody (both free floating and surface bound) possess idiotopes (surface features) to which the receptors of other antibody can bind. As a result of receptor interactions, the repertoire becomes dynamic, where receptors continually both inhibit and excite each other in complex regulatory networks (chains of receptors). The theory suggests that the clonal selection process may be triggered by the idiotopes of other immune cells and molecules in addition to the surface characteristics of pathogen, and that the maturation process applies both to the receptors themselves and the idiotopes which they expose. The immune network theory has interesting resource maintenance and signaling information processing properties. The classical clonal selection and negative selection paradigms integrate the accumulative and filtered learning of the acquired immune system, whereas the immune network theory proposes an additional order of complexity between the cells and molecules under selection. In addition to cells that interact directly with pathogen, there are cells that interact with those reactive cells and with pathogen indirectly, in successive layers such that networks of activity for higher-order structures such as internal images of pathogen (promotion),and regulatory networks (so-called anti-idiotopes and anti-anti-idiotopes).The aim of the immune network process is to make a repertoire of discrete pattern detectors for a given problem domain, where better performing cells suppress low-affinity (similar) cells in the network. This principle is achieved through an interactive process of exposing the population to external information to which it responds with both a clonal selection response and internal meta-dynamics of intra-population responses that stabilizes the responses of the population to the external stimuli.

4) Dendritic Cell Algorithm:

The Dendritic Cell Algorithm is inspired by the Danger Theory of the mammalian immune system, and specifically the role and function of dendritic cells. The Danger Theory was proposed by Matzinger and suggests that the roles of the acquired immune system is to respond to signals of danger, rather than discriminating self from non-self . The theory suggests that antigen presenting cells (such as helper T-cells) activate an alarm signal providing the necessarily co-stimulation of antigen-specific cells to respond. Dendritic cells are a type of cell from the innate immune system that respond to some specific forms of danger signals. There are three main types of dendritic cells: 'immature' that collect parts of the antigen and the signals, 'semi-mature' that are immature cells that internally decide that the local signals represent safe and present the antigen to T-cells resulting in tolerance, and 'mature' cells that internally decide that the local signals represent danger and present the antigen to T-cells resulting in a reactive response. The information processing aim of the algorithm is to prepare a set of mature dendritic cells (prototypes) that provide context specific information about how to classify normal and anomalous input patterns. This is achieved as a system of three asynchronous processes of :

1) migrating sufficiently stimulated immature cells,
2) promoting migrated cells to semi-mature (safe) or mature (danger) status depending on their accumulated response
3) labeling observed patterns as safe or dangerous based on the composition of the sub-population of cells that respond to each pattern.

D. Physical Algorithms

We limit our discussion of physical algorithms only to 3 types of such classes.

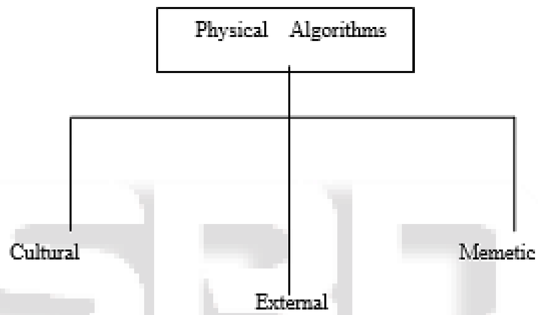

Fig. 5: Classification of Physical Algorithms

1) Memetic Algorithm:

Memetic Algorithms are inspired by the interplay of genetic evolution and memetic evolution. Universal Darwinism is the generalization of genes beyond biological-based systems to any system where discrete units of information can be inherited and be subjected to evolutionary forces of selection and variation. The term 'meme' is used to refer to a piece of discrete cultural information, suggesting at the interplay of genetic and cultural evolution. The genotype is evolved based on the interaction the phenotype has with the environment. This interaction is metered by cultural phenomena that influence the selection mechanisms, and even the pairing and recombination mechanisms. Cultural information is shared between individuals, spreading through the population as memes relative to their fitness or fitness the memes impart to the individuals. Collectively, the interplay of the geneotype and the memeotype strengthen the fitness of population in the environment. The aim of the information processing mechanism is to exploit a population based global search technique to broadly locate good areas of the search space, combined with the repeated usage of a local search heuristic by individual solutions to locate local optimum. Ideally, memetic algorithms embrace the duality of genetic and cultural evolution, allowing the transmission, selection, inheritance, and variation of memes as well as genes.

2) Cultural Algorithm:

The Cultural Algorithm is inspired by the principle of cultural evolution. Culture includes the habits, knowledge, beliefs, customs, and morals of a member of society. Culture does not exist independent of the environment, and can interact with the environment via positive or negative feedback cycles. The study of the interaction of culture in the environment is referred to as Cultural Ecology. The Cultural Algorithm may be explained in the context of the inspiring system. As the evolutionary process unfolds, individuals accumulate information about the world which is communicated to other individuals in the population. Collectively this corpus of information is a knowledge base that members of the population may tap-into and exploit. Positive feedback mechanisms can occur where cultural knowledge indicates useful areas of the environment, information which is passed down between generations, exploited, refined, and adapted as situations change. Additionally, areas of potential hazard may also be communicated through the cultural knowledge base. The information processing aim of the algorithm is to improve the learning or convergence of an embedded search technique (typically an evolutionary algorithm) using a higher-order cultural evolution. The algorithm operates at two levels: a population level and a cultural level. The population level is like an evolutionary search, where individuals represent candidate solutions, are mostly distinct and their characteristics are translated into an aim or cost function in the problem domain. The second level is the knowledge or believe space where information acquired by generations is stored, and which is accessible to the current generation. A communication protocol is used to allow the two spaces to interact and the types of information that can be exchanged.

3) Extremal Algorithm:

Extremal Optimization is inspired by the Bak-Sneppen self-organized criticality model of co-evolution from the field of statistical physics. The self-organized criticality model suggests that some dynamical systems have a critical point as an attractor, whereby the systems exhibit periods of slow movement or accumulation followed by short periods of avalanche or instability. Examples of such systems include land formation, earthquakes, and the dynamics of sand piles. The Bak-Sneppen model considers these dynamics in co-evolutionary systems and in the punctuated equilibrium model, which is described as long periods of status followed by short periods of extinction and large evolutionary change. The dynamics of the system result in the steady improvement of a candidate solution with sudden and large crashes in the quality of the candidate solution. These dynamics allow two main phases of activity in the system:

1) To Exploit Higher Quality Solutions In A Local Search Like Manner,
2) Escape Possible Local Optima With A Population Crash And Explore The Search Space For A New Area Of High Quality Solutions.

The aim of the information processing mechanism is to iteratively identify the worst performing components of a given solution and replace or swap them with other components. This is achieved through the allocation of cost to the components of the solution based on their contribution to the overall cost of the solution in the problem domain. Once components are assessed they can be ranked and the weaker components replaced or switched with a randomly selected component.

E. Stochastic Algorithms:

These are the highly useful stochastic algorithms. We have presented 6 of them which are deployed in 70% of artificial intelligence systems and extensively used in the field of robotics.

1) Random Search:

The mechanism of Random Search is to sample solutions from across the entire search space using a uniform probability distribution. Each future sample is independent of the samples that come before it. Random search is minimal in that it only requires a candidate solution construction routine and a candidate solution evaluation routine, both of which may be calibrated using the approach. The worst case performance for Random Search for locating the optima is worse than an Enumeration of the search domain, given that Random Search has no memory and can blindly resample. Random Search can return a reasonable approximation of the optimal solution within a reasonable time under low problem dimensionality, although the approach does not scale well with problem size (such as the number of dimensions). We should be careful with some problem domains to ensure that random candidate solution construction is unbiased. The results of a Random Search can be used to seed another search technique, like a local search technique (such as the Hill Climbing algorithm) that can be used to locate the best solution in the neighborhood of the 'good' candidate solution.

2) Adaptive Random Search:

The Adaptive Random Search algorithm was designed to address the limitations of the fixed step size in the Localized Random Search algorithm. The mechanism for Adaptive Random Search is to continually approximate the optimal step size required to reach the global optimum in the search space. This is achieved by trialling and adopting smaller or larger step sizes only if they result in an improvement in the search performance. The Mechanism of the Adaptive Step Size Random Search algorithm (the specific technique reviewed) is to trial a larger step in each iteration and adopt the larger step if it results in an improved result. Very large step sizes are trialled in the same manner although with a much lower frequency.

This mechanism of preferring large moves is intended to allow the technique to escape local optima. Smaller step sizes are adopted if no improvement is made for an extended period .Adaptive Random Search was designed for continuous function optimization problem domains. Candidates with equal cost should be considered improvements to allow the algorithm to make progress across plateaus in the response surface. Adaptive Random Search may adapt the search direction in addition to the step size.

3) Stochastic Hill Climbing:

The mechanism of the Stochastic Hill Climbing algorithm is iterate the process of randomly selecting a neighbor for a candidate solution and only accept it if it results in an improvement. The mechanism was proposed to address the limitations of deterministic hill climbing techniques that were likely to get stuck in local optima due to their greedy acceptance of neighboring moves. Stochastic Hill Climbing was designed to be used in discrete domains with explicit neighbors such as combinatorial optimization (compared to continuous function optimization). The algorithm's mechanism may be applied to continuous domains by making use of a step-size to define candidate-solution neighbors (such as Localized Random Search and Fixed Step-Size Random Search). Stochastic Hill Climbing is a local search technique (compared to global search) and may be used to refine a result after the execution of a global search algorithm. Even though the technique uses a stochastic process, it can still get stuck in local optima. Neighbors with better or equal cost should be accepted, allowing the technique to navigate across plateaus in the response surface. The algorithm can be restarted and repeated a number of times after it converges to provide an improved result (called Multiple Restart Hill Climbing). The procedure can be applied to multiple candidate solutions simultaneously, allowing multiple algorithm runs to be performed at the same time (called Parallel Hill Climbing).

4) Guided Local Search:

The mechanism for the Guided Local Search algorithm is to use penalties to encourage a Local Search technique to escape local optima and discover the global optima. A Local Search algorithm is run until it gets stuck in a local optima. The features from the local optima are evaluated and penalized, the results of which are used in an augmented cost function employed by the Local Search procedure. The Local Search is repeated a number of times using the last local optima discovered and the augmented cost function that guides exploration away from solutions with features present in discovered local optima. The Guided Local Search procedure is independent of the Local Search procedure embedded within it. A suitable domain-specific search procedure should be identified and employed. The Guided Local Search procedure may need to be executed for thousands to hundreds-of-thousands of iterations, each iteration of which assumes a run of a Local Search algorithm to convergence.

– The algorithm was designed for discrete optimization problems where a solution is comprised of independently assessable 'features' such as Combinatorial Optimization, although it has been applied to continuous function optimization modeled as binary strings.
– The λ parameter is a scaling factor for feature penalization that must be in the same proportion to the candidate solution costs from the specific problem instance to which the algorithm is being applied. As such, the value for λ must be meaningful when used within the augmented cost function (such as when it is added to a candidate solution cost in minimization and subtracted from a cost in the case of a maximization problem).

5) Greedy Randomized Adaptive Search:

The aim of the Greedy Randomized Adaptive Search Procedure is to repeatedly sample stochastically greedy solutions, and then use a local search procedure to refine them to a local optima. The mechanism of the procedure is centered on the stochastic and greedy step-wise construction mechanism that constrains the selection and order-of-inclusion of the components of a solution based on the value they are expected to provide. The α threshold defines the amount of greediness of the construction mechanism, where values close to 0 may be too greedy, and values close to 1 may be too generalized. As an alternative to using the α threshold, the RCL can be constrained to the top n% of candidate features that may be selected from each construction cycle. The technique was designed for discrete problem classes such as combinatorial optimization problems.

6) Tabu Search:

The aim for the Tabu Search algorithm is to constrain an embedded heuristic from returning to recently visited areas of the search space, referred to as cycling. The mechanism of the approach is to maintain a short term memory of the specific changes of recent moves within the search space and preventing future moves from undoing those changes. Additional intermediate-term memory structures may be introduced to bias moves toward promising areas of the search space, as well as longer-term memory structures that promote a general diversity in the search across the search space.Tabu search was designed to manage an embedded hill climbing heuristic, although may be adapted to manage any neighborhood exploration heuristic.ˆ Tabu search was designed for, and has predominately been applied to discrete domains such as combinatorial optimization problems. Candidates for neighboring moves can be generated deterministically for the entire neighborhood or the neighborhood can be stochastically sampled to a fixed size, trading off efficiency for accuracy. Intermediate-term memory structures can be introduced (complementing the short-term memory) to focus the search on promising areas of the search space, called aspiration criteria. Long-term memory structures can be addressed to implement the short-term memory and also to encourage useful consideration of the broader search space, called diversification. Mechanisms include generating solutions with rarely used components and biasing the generation away from the most frequently used solution components.

III. CONCLUSION

This paper presents a critical survey of clever algorithms. Clever algorithms are used extensively not only in artificial intelligence but also in automation in other real life domains like military, healthcare, manufacturing, mining etc. There are many practical concerns with this clever algorithms like ensuring the correct implementation of the algorithms, issues regarding which software platform to use for implementing such algorithms, how to test the validity of the algorithms, issues to consider while comparing the clever algorithms. These aspects are actively being looked upon.

REFERENCES

[1] R. Battiti. Machine learning methods for parameter tuning in heuristics.In 5th DIMACS Challenge Workshop: Experimental Methodology Day,1996.
[2] R. Battiti and M. Brunato. Handbook of Metaheuristics, chapter Reactive Search Optimization: Learning while Optimizing. Springer Verlag, 2nd edition, 2009.
[3] R. Battiti, M. Brunato, and F. Mascia. Reactive Search and Intelligent Optimization. Springer, 2008.
[4] R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. Technical Report TR-95-052, International Computer Science Institute, Berkeley, CA, 1995.
[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiaim genetic algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation, 6(2):182–197, 2002.

[6] E. Zitzler. Evolutionary Algorithms for Multiaim Optimization: Methods and Applications. PhD thesis, Shaker Verlag, Aachen, Germany,1999.
[7] E. Zitzler, M. Laumanns, and S. Bleuler. Metaheuristics for Multiaim Optimisation, chapter A Tutorial on Evolutionary Multiaim Optimization, pages 3–37. Springer, 2004.