A Holistic Review of Soft Computing Techniques

Philip O. Omolaye, Joseph M. Mom, Gabriel A. Igwue

Department of Electrical and Electronics Engineering, University of Agriculture, Makurdi, Nigeria

Email address: philomolaye@gmail.com (P. O. Omolaye), joe.mom@uam.edu.ng (J. M. Mom), gaigwue@yahoo.com (G. A. Igwue)

Abstract: Due to notable technological convergence that brought about exponential growth in computer world, Soft Computing (SC) has played a vital role with automation capability features to new levels of complex applications. In this research paper, the authors reviewed journals related to the subject matter with the aim of striking a convincing balance between a system that is capable of tolerance to uncertainty, imprecision, approximate reasoning and partial truth to achieve tractability, robustness, economy of communication, high machine intelligence quotient (MIQ), low cost solution and better rapport with reality to conventional techniques. This paper gives an insight on four major consortiums of SC that sprang from the concept of cybernetics, explores and reviews the different techniques, methodologies; application areas and algorithms are formulated to give an idea on how these computing techniques are applied to create intelligent agents to solve a variety of problems. The mechanisms highlighted can serve as an inspiration platform and awareness to new and old researchers that are not or fully grounded in this unique area of research and to create avenue in order to fully embrace the techniques in research communities.

Keywords: Machine Intelligence, Soft Computing, Hard Computing, Hybrid Computing, Neural Network, Fuzzy Logic, Evolutionary Computation, Ant Colony Algorithm

1. Introduction

Artificial Intelligence (AI) is a broad field of study with different meanings to researchers depending on individual perspectives. AI techniques are known to be applied to solve complex and ill-defined problems in which soft computing, computational intelligence and granular computing form some of the major offshoot. McCarthy John [143], based on his perspective, sees AI as “computational intelligence” while Zadeh in [234], [235] claimed that computational intelligence is actually Soft Computing (SC). Irrespective of the meaning attached to it, AI or SC has to do with human intelligence which requires complex and advanced reasoning processes and knowledge in executing diverse applications such as control, forecasting, robotics, pattern recognition, medicine, and optimization, signal processing and industrial applications [1], [29], [38], [224]. In this age, there are several techniques used or applied in solving real world problems such as Hard computing (HC), Soft computing (SC) and Hybrid computing (HyC) with striking or peculiar characteristics and features [47]. HC is also known as the conventional methods that are based on mathematical techniques such as crisp systems, binary logic, numerical analysis and finite element analysis. However, the HC has the characteristics of precision and categorization although imprecision and uncertainties are undesirable properties. The HC is easy to model mathematically but requires a precisely stated analytical model, strictly sequential and produces precise solutions in which stability is highly predictable, deterministic in nature and often requires a lot of computation time [111], [168].

The SC acts as an umbrella or suite of computing techniques in which each of the techniques contributes a dissimilar methodology to address a general problem in its domain in such a way that the principal component methodologies complement each other rather than being competitive in nature. SC comprises wide range of terms, encompassing several techniques but for the purpose of this paper, four (4) out of the numerous techniques are treated with a bias treatment of ant colony optimization (ACO). The four (4) SC consortiums are fuzzy system (FS), artificial neural
networks (ANN), evolutionary computation (EC) and probabilistic reasoning (PR).

In this regard, HyC is the combination of HC and SC which inherit their merits and demerits. However, it is used to get the strength of both techniques and overcome their individual limitations [47]. The HyC is classified into three as sequential (pipelining fashion), auxiliary (subroutine) and embedded (fused totally) hybrid system with glaring properties. A well presented schematic diagram of machine intelligence scheme is shown in figure 1.

Figure 1. Component parts of Machine Intelligence.

2. Soft Computing (SC) and Techniques

SC components form partnership, although it allows parallel computations, yield approximate solutions with high tolerance of imprecision, uncertainty, partial truth, and approximation and dispositionality, incorporates stochasticity, capable of dealing with ambiguous, noisy data and to cap it all, using human mind as the driving model [168]. In SC, the tolerance for imprecision and uncertainty is exploited and applied to real world problems frequently to offer more robust, tractability, lower cost, high machine intelligence quotient (MIQ) and economy of communication than those obtained by most HC mathematical techniques [111]. This is capable of addressing several problems in different domains such as control, data mining, forecasting, modeling, optimization, planning, reliability [136] and also in the areas of application such as banking, energy, food industry, industrial production, logistics, medical industry, polymer extrusion process, software engineering, agricultural and environmental to mention a few [45] - [47]. It consists of several techniques and branches, still developing, new ideas are emerging every day and techniques inspired by the activities of the human brain, laws of nature and the behavior of animals. These techniques have been proven to be efficient in solving various complex problems. In this section, four major SC components are discussed with ACO inclusive. The principal consortium of soft computing (SC) are fuzzy logic (FL), neural network (NN) and probabilistic reasoning (PR), with the latter subsuming evolutionary computing including ant colony optimization (ACO), chaos theory (CT) and components of learning theory as shown in figure 2.

Figure 2. Shows schematic diagram of intersections of members of Soft Computing Techniques.
2.1. Fuzzy Logic System (FLS)

Zadeh, who developed the concept of fuzzy logic (FL), was the first to coin the term SC [232]. In a broad sense, Fuzzy logic (FL) mimics the reasoning of a human expert, preserving information through the use of continuous “interest” or “membership” values until the output is produced [109], [184]. FL has the ability to handle complex decisions by verbalizing the whole approaches and operations with potential algorithm but achieves the tremendous results using processing of inputs of a series of if-then directives and usage of several thresholds. The FL systems use information efficiently from robust to uncertain, missing or corrupted data to propagate till final “defuzzification”. Defuzzification is the final step in which the superposition of multiple rules is resolved to determine the output value for the gate. It is used to encode human expert knowledge/heuristics with common-sense, easily interpreted and constraints naturally enforced with relatively straightforward design and implementation [190]. FL systems are relatively cheap because in most cases, training of data is not usually needed except when used to tune a system and also joint or conditional probability distributions are not required [214] - [217]. The general structure of a fuzzy inference system (FIS) with three component parts is depicted in figure 3.

![Figure 3. General structure of a fuzzy logic system (FLS).](image)

The research works on the Fuzzy Logic and Sets have been underway for over 50 years and is difficult to cover all development aspects of this noble subject. A detailed treatment of the scholars and researchers are found in [231] – [233], [235], [237]. The idea of fuzzy sets is introduced by way of examples which are sets with imprecise amplitudes. Zadeh stated that the “membership” in a fuzzy set is not a matter of affirmation or denial, but rather a matter of degree. The Fuzzy Conditional Statements is known as the fuzzy if-then rules; it is sometimes called fuzzy rules or fuzzy implications. The role of fuzzy conditional statements (fuzzy if-then rules) has a great effect on fuzzy systems. This is because Fuzzy systems mimic human actions and take humanlike decisions by using the knowledge about a target system without knowing the components of the model in question. This is achieved with fuzzy if-then rules. A membership function, \( \mu (x) \) (also called universe) shows the extent to which a value from a domain is included in a fuzzy concept. Membership function (MF) values represent the degree to which an object belongs to a fuzzy set. It is used to define a set and can be represented mathematically as [4]:

\[
\mu_A(x) = 1 \quad \text{if } x \in A, \quad \text{and} \quad (1)
\]
\[
\mu_A(x) = 0 \quad \text{if } x \notin A \quad \text{for all values of } x. \quad (2)
\]

The most commonly used in practice are such as bell-shaped MF, triangular MF, Gaussian MF, two-sided Gaussian MF, pi-shaped MF, product of two sigmoidal MFs, difference between two sigmoidal MFs, and trapezoidal MF. A fuzzy set is hence characterized by its membership function, \( \mu_A(x) \) as:

\[
\mu_A(x): X \to [0, 1] \quad (3)
\]

In the meantime, it indicates the membership grade of these elements in fuzzy set \( A \). Further details of membership functions can be found in [185]. Fuzzy reasoning is also known as approximate reasoning which is an inference procedure or rule that deduces conclusions from a set of fuzzy if-then rules and also identified facts. Fuzzy sets define linguistic variables and hence fuzzy inference rules model the system linguistically. In general, the process of fuzzy reasoning can be divided into four steps namely the degree of compatibility, firing strength, qualified (induced) consequent membership functions and overall output membership functions [4], [125], [231]. Jang [98] gave a good account and in-depth analysis of fuzzy inference systems, because of its multidisciplinary and widely embraced nature, the fuzzy inference system is known by numerous other names, such as fuzzy model [201], fuzzy expert system [109], fuzzy associative memory [30], [118], or simply fuzzy system. The fuzzy inference system (FIS) comprises the knowledge and experience of an expert, widely used and popular computing framework based on the concepts of sets of fuzzy control rules,
fuzzy if-then rules, and fuzzy reasoning. Fuzzy inference system can be classified into three types based on the types of fuzzy reasoning and fuzzy if-then rules employed. The categorized FIS are the Mamdani-type FIS [133]-[134], Tsukamoto-type FIS [213], and Takagi-Sugeno-type FIS [201].

2.2. Neural Network (NN)

Neural Networks (NNs) are the result of academic investigations that use mathematical formulations to model nervous system operations. The resulting techniques are being successfully applied in a variety of everyday business and research applications. Artificial neural networks are, as their name indicates, computational networks which attempt to simulate, in a gross manner, the networks of nerve cell (neurons) of the biological (human or animal) central nervous system. In recent years, neural networks (NNs) have been studied very extensively. The brain basically learns from experience. In contrast to traditional mathematical models, which are programmed, NNs learn the relations between selected inputs and outputs such that it can accommodate many inputs in parallel and encode the information in a distributed fashion. Artificial Neural Networks is no more than an interconnection of artificial neurons. NNs learn the relations between selected inputs and outputs from the previous experiences. NNs also perform their tasks simultaneously (i.e. parallel processing), which makes NNs very fast. A typical NN can identify and learn the relationships between the inputs and the outputs of a non-linear multi-dimensional system.

Neural Networks have turned out to become a technical folk legend in which the market is flooded with new, increasingly technical software and hardware products, and many more to come. There are many papers describing applications of the popular implementation of NNs to solve problems in telecom fraud detection [7], Faults Detection in Power Systems [209], Crime detection and management [3], Cellular Networks [150] Aiding in Medical Diagnosis [162]. The primary information processing structures of interest in neuro-computing are Neural Networks (NN). The potential of artificial neural network relies on massively parallel architecture composed of large but finite number of artificial neurons which act as simple computational elements connected by edges with variable weights [82]. There are various models of neural networks which have been reported in literature with trend setting models of neural networks such Perception [16], [177], Adaptive Neural Network [80], [130], Linear Associator Model [32], Little and Shaw Model [95], Hopfield Model [194], Grossberg Models [74], Self-Organizing Network [140], [150] and Back Propagation Network [81]. As a result of the ever-increasing existence of all these networks, the application of ANN is increasing tremendously in different areas of interest and research. The figure 5 shows the ANN taxonomy which showcases the feed-forward and the recurrent feedback network architectures.

The arrangement or topology of neurons into layers and the pattern of connection within and in-between layers are generally known as the architecture of the network. When combining two or more artificial neurons we get an artificial neural network. While single artificial neuron has almost no usefulness in solving real-life problems, the artificial neural networks have. In fact artificial neural networks are capable of solving complex real-life problems by processing information in their basic building blocks (artificial neurons). Mainly, ANNs come in many different shapes and sizes and are classified into two groups depending on the structure of the connections, feed-forward (allowed to pass in one direction, i.e. from input to output) and recurrent (feed-back) networks. A recurrent neural network (RNN) is a class or type of artificial neural network where connections between units form a directed cycle. When all units in all layers are interconnected together, it is called fully connected which is the most general case of the recurrent networks. In recurrent networks, some of the connections may be absent, but there are feedback connections. Therefore, recurrent networks need to be operated over time. This creates an internal state of the network which allows it to exhibit dynamic temporal behaviour. Recurrent artificial neural networks can use their internal memory to process any sequence of inputs.
Figure 6. Fully recurrent artificial neural network.

Figure 6 shows small fully recurrent artificial neural network and complexity of its interconnections. The most basic topology of recurrent artificial neural network is fully recurrent artificial network where every basic building block (artificial neuron) is directly connected to every other basic building block in all directions. Other recurrent artificial neural networks such as Hopfield, Elman, Jordan, bi-directional and other networks are just special cases of recurrent artificial neural networks. Artificial neural network with feed-forward topology is called Feed-Forward artificial neural network and as such has only one condition: information must flow from input to output in only one direction with no back-loops. There are no limitations on number of layers, type of transfer function used in individual artificial neuron or number of connections between individual artificial neurons. If the interconnection matrix is restricted to just one direction (neither feedback nor self connections), the neural network is defined as feed forward. A special group of feed forward networks is layered, which is called the multi-layer perceptron (MLP).

Layered ANNs, basically consist of a set of units that constitute the input layer, one (or more) hidden layer(s) and an output layer. No processing is done in the input layer and because of that its components are called input nodes. This information is used by the neural network to solve a problem. The method which involves the setting of values for the weights enables the process of learning or training [53]. Figure 8 indicates a simple neural network in which the weights are denoted by $w_1$ and $w_2$.

Figure 8. A simple Neural Set.

In the neural network illustrated in Figure 8, the activation of neuron 1 and activation of neuron 2 are $x_1$ and $x_2$ respectively. The output neuron $y$ is calculated by the formula:

$$\text{Net} = x_1w_1 + x_2w_2$$

The act or process of modifying the weights in the connections between the network layers with the aim of achieving the expected result (output) is known as training a network.
network [180] while the internal arrangement or process that happens when a network is trained is called learning [17]. Kayri and Çokluk [112] classified training into supervised, unsupervised and reinforcement. The activation function is used to calculate the output response of neuron in which the sum of the weighted input signal is applied with an activation to obtain the response. Different activation or transfer functions have been proposed by different scholars and researchers satisfactory results such as Lorentzian transfer functions [128], Max-Piecwise-Linear (MPWL) Neural Network for function approximation [221], non-polynomial activation functions [33], Hermite Polynomial [132], Gaussian bars [85], hybridization of various functions such as polynomial, periodic, sigmoidal and Gaussian functions [159], two new activation functions labelled sincos and sinc [63], Hybridization of used complementary log-log and probit functions [78]. Giraud in [77] proposed a new class of sigmoidal functions which has proven satisfied with universal approximation theorem requirement. Also another set of new sigmoidal activation function was proposed for modelling of dynamic, discrete time systems with satisfactory results [195]. Only a few of these are used by default; the others are available for customization discussed in [189].

2.3. Evolutionary Computation (EC)

The field of evolutionary computation was before the formative days of the electronic computer, this is because EC is itself an evolving community of people, ideas, and applications which can be traced to its genealogical roots as far back as the 1930s. The emergence of digital computing technology acts as catalyst of exposure to the world. The fusion of evolution strategies, evolutionary programming, and genetic algorithms had formed the backbone of the field of evolutionary computation which remains an active research area for both old and new researchers [50]. Figure 9 shows the Evolutionary computation consortium that forms the backbone for evolutionary systems. In Between 1950s and the 1960s, various computer scientists and researchers carried out various studies independently on evolutionary systems with the idea to apply operators inspired by natural genetic variation and natural selection as an optimization tool for problem solving techniques in industrial and engineering domain. In mid 1960s, evolution strategies (ESs) were introduced by Rechenberg and Schwefel [181]. This is a technique applied to optimize real−valued parameters for devices such as airfoils. The idea was further fueled by Schwefel which made the field to remain an active area of research and a further review of (ES) could be found in [153], [172]. A technique in which the solutions to given tasks were represented as finite−state machines, evolved by randomly mutating their state−transition diagrams and selecting the fittest was developed by Fogel, Owens, and Walsh [73] and named evolutionary programming (EP). The EP remains an area of active research which can be found in [203]. In the same vein in mid 1960s, Holland invented Genetic algorithms (GAs) with initial idea to formally study the phenomenon of adaptation as it happens in nature and to develop ways in which the mechanisms of natural adaptation might be imported into computer systems, but the idea received a full blast as an algorithm to solve specific problems [166], [171], [186], [212].

![Evolutionary Computation Consortium](image_url)

**Figure 9.** Evolutionary Computation Consortium.

2.3.1. Evolution Strategies (ES)

Rechenberg and Schwefel invented evolution strategies (ESs) in the 1960s based on independent work of Holland (1975) on GAs at the same time [181]. ES is known with distinguished feature of self-adaptation of additional strategy parameters, which makes the technique to adapt the evolutionary optimization process to the structure of the fitness landscape. ESs techniques share common characteristics of EC in the areas of population of genetic structures, selection to exploit inspiring regions of the search space, recombination and mutation to generate new variants. Also, the major difference between ESs and GA is the genetic representation of candidate solutions, where the chromosome in an ES consists of a vector of real numbers and that of GAs deals with binary strings in population processes. The two vectors of real number as being operated by ESs and the additional parameters, $\sigma_i$ define the standard deviation of random mutations used for the corresponding object variable, $x_i$, are represented mathematically in eqn. (5) and eqn. (6) respectively as

$$\bar{x} = \mathbb{R}^n$$ (5)

$$\bar{\sigma} = \mathbb{R}^n_+$$ (6)
In ESs, mutation plays a central role while mutation is considered to be a secondary operator merely useful to prevent premature convergence of the population in GAs. Therefore, mutation is the combination of object variable $x_i$ and the normal distribution of noise $N_i(0, \sigma_i^2)$ with the variance $\sigma_i^2$ as shown in eqn. 7

$$x'_i = x_i + N_i(0, \sigma_i^2)$$

(7)

The parameters, $\sigma_i$ are mutated using a logarithmic normal distribution in which the global factor $e^{(\sigma_i \cdot N(0,1))}$ tends to increase or decrease over all mutability.

$$\sigma'_i = \sigma_i e^{(\sigma_i \cdot N(0,1) \cdot e^{-e \cdot N(0,1)})}$$

(8)

### 2.3.2. Genetic Programming (GP)

Koza in [119] invented Genetic programming (GP) by extending the capability of genetic algorithm which uses the ideas and terms of biological evolution to handle a complex problem with a model for testing and selecting the best choice among a set of results, represented by a string. The term Genetic Programming (GP) as stated by [223], describes it as a research area within the field of Evolutionary Computation that deals with the evolution of computer code. GP is the automatic creation of computer programs to perform a selected task using Darwinian natural selection. GP is a dynamic, robust and quickly growing discipline with high visibility as a promising research area and remarkable successes are found in different domain such as electronic circuit design, data mining, concrete estimation, robot control, optimization and pattern recognition in [11], [39], [119], [129], [160]. A most difficult exercise in using genetic programming to perform task is by determining the fitness function and the degree or rate at which the program is helping to arrive at the desired goal. It is capable of cross-breeding with other programs such as list processing (LISP), Scheme as well as C and other programming languages to continually approach the needed solution. However, it can also be used with C and other programming languages.

### 2.3.3. Evolutionary Programming (EP)

Evolutionary programming (EP) originated from the work of L.J. Fogel in the 60s [72] and was further extended by his son D.B. Fogel in the late 80s [69]. The 1966 book, "Artificial Intelligence through Simulated Evolution" by Fogel, Owens and Walsh was the landmark for EP applications, although many other journals papers appeared earlier in the literature but impact felt was minimal [73]. In its current status, EP shares a number of characteristic features with ES, such as a real-number representation, normally distributed mutations and adaptation of mutation variances. EP is considerably applicable to any of the domain for which evolutionary algorithms are applicable and has been applied to diverse engineering domain including traffic routing and planning [100], data mining [166], pharmaceutical design [168], cancer detection [6], [70], military planning and learning[71], [93], [210], control and power system [9] among others. EP applies the scaling function $\delta$ to map objective function values $f(\vec{x})$ to fitness values and EP mutation is shown in eqn. 9

$$x'_i = x_i + \sqrt{\beta_i \cdot \Phi(\vec{a}) + \gamma_i \cdot N_i(0,1)}$$

(9)

where

$$\Phi(\vec{a}) = \delta(f(\vec{x}))$$

(10)

The constants $\beta_i$ and $\gamma$ are the two exogenous parameters that must be manually adapted to a particular optimization assignment.

### 2.3.3. Genetic Algorithm (GA)

A genetic algorithm (GA) is a heuristic optimization method [147], [171], [219] biologically inspired approximate solutions for optimization and search technique. It was proposed and developed by Holland [91] in which its behavior mimics the evolution of simple, single celled organisms and the algorithm works with binary strings to yield a secondary generation of strings through the genetic operators of crossover and mutation. The cycle is repeated over and over again until a termination condition is reached and those that survive the evolution process are chosen according to the fitness of chromosomes. For clear and better understanding of GA concept, having an understanding background of biological evolution is required since in nature and ideal situation, competition among individuals for resources such as space and food leads to the domination of strong individuals over the weaker ones. Only the fittest can survive and reproduce in such an environment hence, the genes of the fittest survive [15], [198]. The three most common terminologies used in GA are Gene which acts as the entity that represents a particular characteristic of individual; Chromosome comprises a string or collection of genes which represents an encoding of solution and; Population is the collection of chromosomes. The relationships are shown in figure 10. The gene in GA takes on a value from an alphabet of size $\in \mathbb{Z}^+$ from each string or chromosome then consists of a series of $n$ genes resulting in a solution space of size $\mathbb{R}^n$. The binary alphabet $\{0, 1\}$ is the most common and generally most effective string which means each gene takes on a possible value of 0 or 1:

$$b_i \in \{0, 1\}, \quad i = 1,2,\ldots,n$$

(11)

$$\vec{s} = \{b_1, b_2, \ldots, b_n\}$$

(12)

![Figure 10. Genetic Algorithm Systems.](image-url)
GA has established itself to be extremely effective in problems such as software testing [167], [170], medical diagnosis [68], [121], airline booking [13], seismic vibration [161], mechatronics and robot [76] and load-balancing [238], networks [110]. The simple algorithm of GA and flow chart for chromosome population are shown in algorithm 2 and figure 11 respectively.

Algorithm 1 (GA for chromosome population):
Input: GA Parameters.
Output: Population of chromosomes, $P$. Each of these chromosomes represents a solution to the problem.

```
Begin
  Initialize the population, $P$;
  Evaluate or select population, $P$;
  While stop or terminate conditions not true or met do
    Apply Selection method in $P$ to create next population $P_n$;
    Perform Crossover $P_n$;
    Perform Mutate $P_n$;
    Replace $P$ with $P_n$;
    Evaluate population, $P$;
  End.
```

Figure 11. Flow chart of fundamental mechanism of simple genetic algorithm.

2.4. Probabilistic Reasoning (PR)

The variant of PR is probability logic and Bayesian (also known as Belief). Networks are a powerful knowledge representation and reasoning mechanism used in PR. Uncertainty (such as missing data, uncertain inputs, noisy data, incompleteness of knowledge, imprecision, variability of the described phenomena, approximations made for the sake of simplicity) hardly need any introduction because they are perhaps the most inherent and the most prevalent characteristics of the knowledge of the world around us. However, the probabilistic reasoning involves both the probability theory and the methodology by which we choose specific probability models and apply them since the concept requires a stand on its meaning and domain of applicability (uncertainty, chance, indeterminacy). Uncertainty in probability theory is measured by a real number between Zeros (0) of impossible event and Ones (1) of sure event. Some major formalisms for representing and reasoning about uncertainty are Mycin’s certainty factors (an early representative), probability theory (esp. Bayesian belief networks), Dempster-Dhafer theory, fuzzy logic, truth maintenance systems, non-monotonic reasoning. The major concept of a probabilistic is to fuse the capacity of probability theory to handle uncertainty with the capacity of deductive logic to exploit structure. The PR can be considered as an analogy to fuzzy reasoning by taking into consideration the uncertainty of fuzziness as related to the concept of approximation. A comprehensive research work on the subject matter is found in [51], [220], and [229]

2.5. Ant Colony Algorithm (ACA)

ACA inspired by the behavior of real ant colonies observations, was first introduced by [41] as a kind of distributed optimization tool which was applied to solve the
traveling salesman problem (TSP) [56], [58]. ACA has recorded huge success when used to solve other combinatorial problems including graph coloring [42], data mining and optimization [59], [160], Machine learning [48], [49], [69], [141], [158], [165] mobile robots [31], [155], network routing and load-balancing [193], vehicle routing [55], fuzzy controller [108], quadratic assignment [135] and the shortest common super-sequence problem [145]. Although, a global behavior emerges, through these simple local interactions between individual ants since each of them possesses only limited skill and communication capability. In order to survive, ants require finding the shortest route or path to leave their nest and forage for food [24]. That is where the slogan of the survival-of-the-fittest mechanism comes in, which entails that ants do not only find food, but also find an optimal path or shortest route between their nest and the food source in [84], [89], [90], [92], [176], [200]. Different species and colonies of ants with different mode of searching for foods and the whole foraging process of ants may be grouped and described by three strategies: hunting, homing, and path building which has captured the imaginations of many researchers in [24], [27], [60], [124], [127], [163], [175] and [199]. There are two types of ACA that can be implemented, type-1 and type-2. The type-1 algorithm framework is shown in Algorithm 2.

Algorithm 2 (ACA for cell planning and reporting):
Input: ACA Parameters
Output: Solutions representing suitable reporting cell form factors.
Begin
Initialize population and pheromone.
While stopping conditions not true do
For each ant:
Construct or Generate solution using the current pheromone trail (iteration)
Evaluate the generated solution (pheromone)
Update pheromone trail
If escape condition true then Escape.
End

3. Comparison of Independent Techniques

A detailed comparison of different intelligence systems of fuzzy system (FS), neural network (NN), evolutionary computation (EC), probabilistic reasoning (PR) with classical approach of artificial intelligence (AI) is given in Table 1 using conventional Fuzzy terms for grading systems such as good, slightly good, slightly bad and bad. The comparison was based on very meticulous analysis and influential predictors known Key Performance Indicators (KPIs). The major indices called parameters used as shown on Table 1, which can be ranked or grouped, because they are closely associated with each other.

<table>
<thead>
<tr>
<th>S/N</th>
<th>PARAMETERS</th>
<th>FS</th>
<th>NN</th>
<th>EC</th>
<th>PL</th>
<th>AI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Adaptability</td>
<td>Bad</td>
<td>Good</td>
<td>Slightly bad</td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td>2</td>
<td>Expert Knowledge representation</td>
<td>Good</td>
<td>Bad</td>
<td>Slightly bad</td>
<td>Slightly good</td>
<td>Good</td>
</tr>
<tr>
<td>3</td>
<td>Explanation ability</td>
<td>Good</td>
<td>Bad</td>
<td>Slightly bad</td>
<td>Slightly good</td>
<td>Good</td>
</tr>
<tr>
<td>4</td>
<td>Fault tolerance</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>5</td>
<td>Impression tolerance</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>6</td>
<td>Learning capability</td>
<td>Bad</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>7</td>
<td>Maintainability</td>
<td>Slightly good</td>
<td>Good</td>
<td>Slightly good</td>
<td>Slightly good</td>
<td>Slightly good</td>
</tr>
<tr>
<td>8</td>
<td>Mathematical model</td>
<td>Slightly good</td>
<td>Bad</td>
<td>Bad</td>
<td>Good</td>
<td>Slightly bad</td>
</tr>
<tr>
<td>9</td>
<td>Non-linearity</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Slightly bad</td>
</tr>
<tr>
<td>10</td>
<td>Optimization ability</td>
<td>Bad</td>
<td>Slightly good</td>
<td>Good</td>
<td>Slightly good</td>
<td>Bad</td>
</tr>
<tr>
<td>11</td>
<td>Real-time operation</td>
<td>Good</td>
<td>Slightly good</td>
<td>Slightly bad</td>
<td>Good</td>
<td>Bad</td>
</tr>
<tr>
<td>12</td>
<td>Uncertainty tolerance</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Good</td>
<td>Bad</td>
</tr>
</tbody>
</table>

4. Applications of Soft Computing

Soft computing which has become one of the promising areas of research has two main advantages: firstly in the area of solving non-linear problems where mathematical techniques are not feasible or available and secondly by providing reasonable solution through human knowledge in form of cognition, understanding, recognition, learning, and other related entity. SC techniques are employed in different fields resulting in construction of intelligent systems in the likes of autonomous self-tuning and automated designed systems as analyses and harmonized in table 1. This is down to the fact that approach applied a combination or fusion of different decision making models, knowledge representation schemes and learning strategies to solve a computational task. The mark sign of “√” is used to show where there exist tapped areas and the null shows that there is need for more attention from researchers.

<table>
<thead>
<tr>
<th>S/N</th>
<th>FIELD OF APPLICATIONS</th>
<th>SC Components</th>
<th>GA</th>
<th>ACA</th>
<th>REFERENCES by researchers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Aircraft, booking and spacecraft</td>
<td>√</td>
<td>√</td>
<td>✔</td>
<td>[67], [75], [97], [113], [178], [197], [221], [222]</td>
</tr>
<tr>
<td>2</td>
<td>Antenna design</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>[101], [102]</td>
</tr>
<tr>
<td>3</td>
<td>Communication networks</td>
<td>√</td>
<td>√</td>
<td>✔</td>
<td>[79], [139], [152], [154], [156], [173], [208]</td>
</tr>
</tbody>
</table>
5. Conclusion

SC techniques have already widened beyond its current constituents, by making it a major force to reckon with in the area of educational and industrial research. The paper concludes by given analyzing, significant review and justification of soft computing methods by focusing on prime contribution offered by its consortiums, diverse features with sufficient up to date references for the convenience of readers. This is because it has taken a drastic and exponential dimension in growth day by day in many fields of study. The major components that form soft computing and their applications were thoroughly reviewed, the experimental performance in terms of broad embracement in virtually all sectors over the conventional method were explored and established.

The performances of the techniques were analyzed and compared independently by putting into consideration different parameters as shown in table 1. Since the technique has played vital role for finding lasting solution to real life problems very efficiently and cost effectively, we made it a point of duty to demystify the techniques in order to encourage both old and new researchers to understand the concepts, and embrace the current trends and further assist in research directions and prospects.

The theories and applications of soft computing have shown that some of the consortium is in full swing while some are still relatively new field of research. Therefore, the incredible amount of researches conducted on soft computing techniques and numerous proposed methods still have a lot of potential for advancement and improvement, it is worth noting and hoped that this holistic review will add value to researchers, who are keen to further in contributing to the field hybridization of SC.

6. Future Research and Direction

According to research conducted so far, it was noticed that some interesting research area has not been fully tapped which need to attract additional researchers at undergraduate, master’s and doctorate level. We hope that the holistic review and application areas will arouse the kin interest of some researchers and technological communities in prosperous and challenging research field.

References


Wolfgang, B. (2012). Evolutionary Computation and Genetic Programming, Department of Computer Science, Memorial University of Newfoundland, St. John’s, A1B 3X5, CANADA.


