IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 4 ISSUE : 12 Print / Issue Publication Date: 10-Jun-2020

ISSN : 2455-2143

DOI : 10.33564/IJEAST.2020.v04i12.006

Indexed In

Google Scholar
OPEN ACCESS
doi
Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 4, Issue 12, ISSN No. 2455-2143, Pages 36-49

SURVEY: BIO-INSPIRED ALGORITHMS IN THE HUMAN BODY SYSTEM

Baida'a Lala'a
Computer Science
Faculty of Computer and
Information Technology,
Sana'a University, Sana'a, Yemen

Ghaleb Al-Gaphari
Computer Science
Faculty of Computer and
Information Technology,
Sana'a University, Sana'a, Yemen

Nabeel Alsohybe
Information Technology
Management
Faculty of Computer and
Information Technology,
Sana'a University, Sana'a, Yemen

Abstract— **Optimization algorithms are techniques for solving problems in different areas intelligently by choosing the best solution among available solutions. One of these techniques is metaheuristic. Metaheuristics are scientific techniques that have been inspired by nature for solving optimization problems. They have good properties such as simplicity, flexibility, and high efficiency. Since the human body is made up of multifunctional systems that cover different areas that can be inspired to be employed in solving optimization problems. However, there is a lack in terms of survey research papers related to optimization techniques, which inspired by the human body system. Therefore, this paper introduces a literature review covering the fundamentals and some applications of almost all metaheuristic algorithms inspired by the human body system, which previously have been studied. Also, this paper compares these algorithms in terms of their solution methodologies. It aims to guide researchers in this field to create a general method that inspired by specific human body organs, which may efficiently solve a wide range of tough optimization problems with high accuracy.**

Keywords— **Human Body System Inspired Algorithms (HBSIAs), Metaheuristic, Inspired-algorithms, Human body system algorithms.**

I. INTRODUCTION

Human thinking has been developed through human experience during life. That experience comes from situations, facts, information, or meditating of divine systems behavior, which guides people to think about a set of solutions then choose the best solution among them. In the real world, people are always trying to optimize something, whether to minimize (path, cost, response time, energy consumption…etc.) or to maximize (profit, output, performance, efficiency…etc.). Since, resources, time, and money are always limited; consequently, optimization is the far more (X.-S. Yang et al. 2013). Someone cannot obtain an optimal precise and complete solution as those existed in the universe, whether nature behavior in living creatures, cosmic systems, or at the level of the mechanism of systems in the human body. According to (X.-S. Yang et al. 2013), One of the most effective ways to solve optimization problems is metaheuristic algorithms (Bio-inspired computation methods). Bio-inspired computation methods have been inspired by the behavior of biological systems and physical systems in nature. An example of the behavior of biological systems is particle swarm optimization, which was developed based on the swarm behavior of birds, fish, and insects (Kennedy and Eberhart 1995; Karaboga 2005; Karaboga and Basturk 2007; X.-S. Yang and Deb 2009). While simulated annealing (Kirkpatrick, Gelatt, and Vecchi 1983) is an example of physical systems inspired methods, which was based on the annealing process of metals. Bio-inspired algorithms have spread into different areas of sciences, engineering, and industries. Therefore, it is perhaps one of the most active and popular research subjects with vast multidisciplinary connections. As a result, most researches in the field of optimization have turned to bio-inspired computation (X.-S. Yang et al. 2013). The time sequence of bio-inspired algorithms has been sped up in the last 20 years (Hussain et al. 2019). Table 1 shows some successful metaheuristic algorithms (Hussain et al. 2019), and the targeted algorithms in this paper.

This paper attempts to introduce almost all metaheuristic algorithms which inspired by the human body system. Then, it will compare HBSIAs solution methodologies with an attempt to prove their success in real life by showing some of their applications in solving optimization problems.

The rest of this paper is organized as follows: Section II. discusses Human Body System Inspired Algorithms (HBSIAs). Then Section III compares between HBSIAs. It followed by Section IV, which presents HBSIAs Applications. Finally, the conclusion in Section V.

Algorithms
Artificial Neural Network (McCulloch and Pitts 1943)
Genetic algorithm (1975)
Simulated Annealing (Kirkpatrick, Gelatt, and Vecchi 1983)
Tabu search (Glover 1989)
Artificial Immune System (Farmer, Packard, and Perelson 1986)
Ant colony optimization (Colorni, Dorigo, and Maniezzo 1992)
Particle Swarm Optimization (PSO)(Kennedy and Eberhart 1995)
Variable neighborhood search (VNS) (Mladenović and Hansen 1997)
Harmony search (Geem, Kim, and Loganathan 2001)
Artificial Bee Colony (Karaboga 2005)
Firefly Algorithm (X.-S. Yang 2008)
Cuckoo Search (X.-S. Yang and Deb 2009)
Bat Algorithm (X.-S. Yang 2010a)
Fireworks algorithm (FWA)(Tan and Zhu 2010)
Heart Algorithm (Hatamlou 2014)
Differential search (Sulaiman et al. 2014)
Kidney Algorithm (Jaddi, Alvankarian, and Abdullah 2017)
Artificial Coronary Circulation System (Kaveh and Kooshkebaghi 2019a)

Table 1: Metaheuristic Algorithms Examples

II. HUMAN BODY SYSTEM INSPIRED/ METAHEURISTIC ALGORITHMS

A. Metaheuristics

Heuristic algorithms are algorithms that use a trial-and-error approach for generating new solutions while metaheuristic algorithms are a higher-level heuristic with the use of memory, solution history, and some other forms of learning strategy (X.-S. Yang 2018). In contrast with traditional algorithms, metaheuristics have the following advantages and characteristics:

- They often inspired by nature.
- Metaheuristic algorithms are approximate and usually using stochastic components. Thus, no identical solutions can be obtained, even if it begins with the same initial points. The final solutions, however, can be sufficient close, and they often enable the algorithm to escape any local modes(X.-S. Yang 2018).
- A metaheuristic is not problem-specific and often treat problems without specific knowledge. Therefore, metaheuristic has several parameters that need to be set up associating with optimization problems; consequently, they can solve a broader range of problems.
- The goal of metaheuristic algorithms is to explore the search space for finding near-optimal solutions efficiently.

B. Human body system inspired algorithms

The human body is made up of a set of multifunctional systems that integrate to keep human body systems systematically. These systems covered different areas that can be inspired to use them in technical fields. Figure 1 shows the classification of some general functions in the human body system.

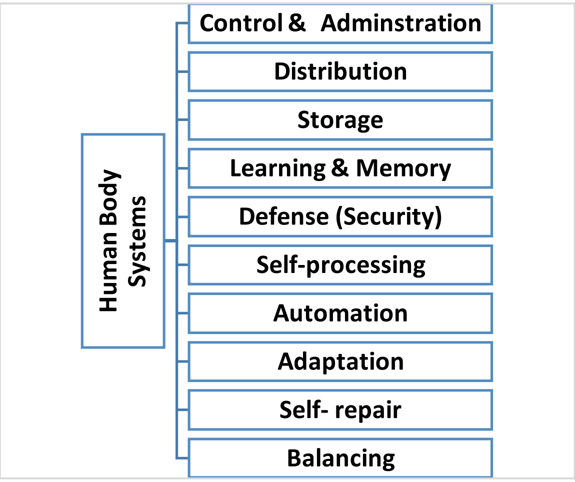

Figure 1: Functions of the human body systems

Since these systems are accurate, algorithms on the technical side have been inspired by these systems, which have been proven to be very successful. This has led to use them in modern technological fields (as shown in section IV).
These algorithms include Artificial Neural Network (ANN), Genetic Algorithm (GA), Tabu Search (TS), An Artificial immune system (AIS), Heart algorithm (HA), Kidney algorithm (KA), and Artificial Coronary Circulation System (ACCS).

1. Artificial Neural Network (ANN)

McCulloch and Walter Pitts (McCulloch and Pitts 1943) created a computational model for neural networks. ANN is a model that imitates the network of interconnected neurons with the neural activities in human brains and solves problems in the same way that a human brain would. It is generally believed that the main computational component of the brain is the neuron. There are almost 86 billion neurons in the average human brain (Sze et al. 2017). The neurons themselves are connected with several elements entering them called dendrites, and an element leaving them called an axon, as shown in Figure 2. The neuron receives the signals entering it via the dendrites, executes a computation on those signals, then outputs this signal on the axon. These input and output signals are called activations of neurons. The axon of one neuron branches out, and it is connected to the dendrites of many other neurons. Synapse is the connections between a branch of the axon and a dendrite.

ANN uses *connectionism*, *interactions*, *memory, and learning* (X.-S. Yang et al. 2013) as a method to create solutions.

Connectionism: ANN is based on a collection of connected units or nodes called artificial neurons. Each connection, such as the synapses in a biological brain, can transfer a signal to other neurons. An artificial neuron receives a signal, then handles it, and can also signal neurons connected to it.

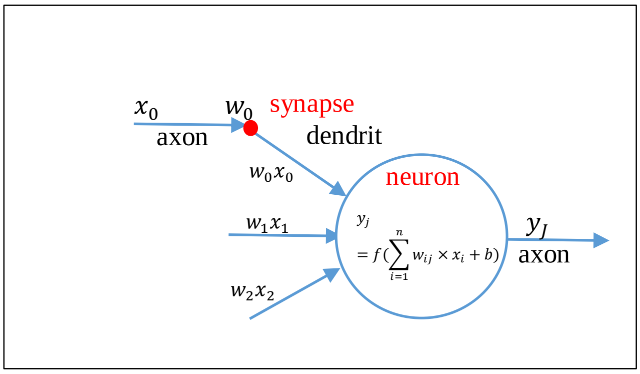

Figure 2: Connections to a neuron in the brain; $\boldsymbol{x_i}$, $\boldsymbol{w_{ij}}$, $\boldsymbol{f}$ (.), and $\boldsymbol{b}$ are the activations, weights, non-linear function, and bias, respectively **(Sze et al. 2017)**.

Interactions: The neurons are typically organized into multiple layers, especially in deep learning. Neurons exist in one layer connect only to neurons of the immediately preceding and immediately following layers. In general, ANN contains three main layers:

- *Input layer:* Is the layer that receives external data.
- *Output layer*: Is the layer that produces the final result.
- *Hidden layer:* In between input and output layers are zero or more hidden layers. In this layer, training and learning phases are done to produce the ultimate results. There are two forms of learning, supervised and unsupervised learning. In supervised learning, the network is trained to get expected output by dynamically updating its weight vector (Training phase - input and output are known). While in unsupervised learning, a network receives input and tries to learn about the input distribution (Learning phase - conclusion of the output).

Memory: Long Short-Term Memory networks (LSTMs) are a popular variant that has internal memory to allow long-term dependencies to affect the output. In these networks, some interfere operations generate values that are stored internally in the network to use them as inputs to other activities in the processing of a later input.

Learning: is the network adaptation to better handle a task by considering sample observations. Learning involves adjusting the weights of the network by minimizing the observed errors to improve the accuracy of the result. The basic program does not change as it learns to execute its given tasks. However, in specific cases of ANNs, this learning involves determining the value of weights and is referred to as "training the network.". When trained, the program in ANN can achieve its tasks by computing the output of the network using the weights that determined during the training phase. Inference occurs when the program is running with these weights. Most learning models can be considered as a simple application of optimization theory and estimation.

2. Genetic Algorithm (GA)

Holland proposed a nature-inspired algorithm named a Genetic Algorithm(1975). GA is a mimic search method inspired by the abstraction of Darwinian evaluation and the natural selection of biological systems. GA is summarized into the genetic operators: crossover or recombination, mutation, fitness, and the selection of the fittest (X.-S. Yang 2010b). Genetic Algorithm can be summarized into the following steps:

a. Initialization.

An individual in the population in GA is a string of symbols and is an abstract representation of the solution. The symbols are called genes, and every string of genes is called a chromosome (Figure 3). In the GA, the initial population of candidate solutions is usually generated randomly across the search space. This population is represented by the parallel search, which achieved by keeping a set of possible solutions for the optimization problem.

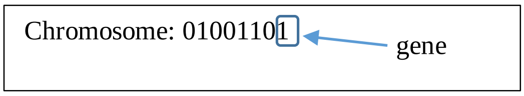

Figure 3: Example of an individual in the population.

b. Evaluation.

Once the population is initialized, or an offspring (children) population is created. Some fitness measures evaluate the individuals in the population.

c. Selection.

After the evaluation process is completed. The survival-of-the-fittest mechanism chooses more copies of the candidate solutions with higher fitness values

d. Crossover

Crossover is obtaining new individuals by combining genetic material from two selected parent chromosomes, possibly better solutions or not. There are many ways to achieve crossover operations, and competent performance depends on a properly designed recombination mechanism. The offspring under crossover will not be identical to any particular parent.

e. Mutation.

While crossover operates on two or more parental chromosomes, the mutation is unary operators that alter the genetic structure of a single chromosome. It locally but randomly flips some bits in a chromosome (modifies a solution).

f. Replacement. The offspring population created by genetic operators (selection, crossover, and mutation).

Offspring replaces the original parental population to start a new cycle life of reproduction to access to an optimal solution.

g. Repeat steps b–g until a terminating condition is satisfied. Figure 4 shows the GA cycle of reproduction.

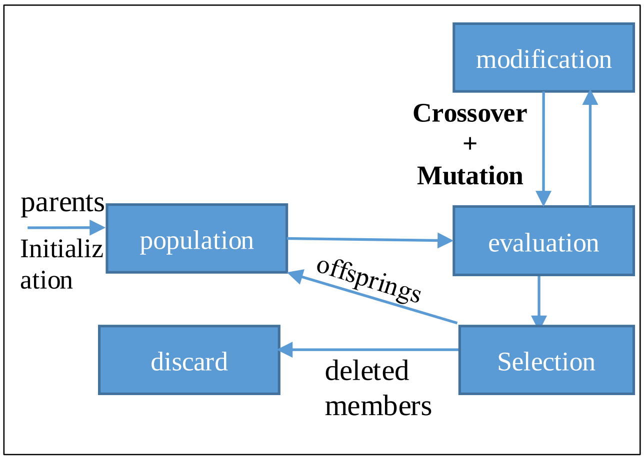

Figure 4: The GA Cycle of Reproduction

3. Tabu Search (TS)

Glover et al. proposed TS (Glover 1986). TS is inspired by the mechanisms of human memory. In TS, a list or memory has been created, and it is called tabu list, which starts to store searched better distinct moves and does not allow to coming back at the already searched move. Thus, it prevents endlessly repeated cycling of the same solution search. Also, within this process, it allows accepting not better move as compared to repeated move. The key feature of TS is the length of the tabu list. Short length permits local search while long length permits global search.

TS has been summarized into the following steps:

- Initialization phase: Begin within a starting current solution (from intermediate or long term memory).
- Create a candidate list of moves (each move would generate a new solution).
- Choose the best allowable candidate based on tabu restriction and aspiration criteria.
- Stop if stopping criteria are satisfied, or transfer. Transfer initiates diversification or intensification phase embodied in intermediate or long term memory to select the best allowable candidate.

The memory structures in TS operate by reference to four principal dimensions: *Recency, Frequency, Quality, and Influence.*

a. Recency-based memory keeps track of solutions attributes that have varied during the recent past. To exploit the memory in TS, selected attributes that exist in solutions recently visited are called tabu-active, and solutions that have tabu-active elements, or specific collections of these attributes, are those that become tabu. Therefore, this prevents specific solutions from the recent past from being revisited. Other solutions that share like tabu-active attributes are also similarly prevented from being visited. Tabu classification strictly refers to solutions which are forbidden to be visited by breaking specific restriction based on these attributes.

b. Frequency-based memory refines representation by accounting for the number of times each edge appears in the critical solutions and allows the inclusion of additional weighting factors.

c. The quality refers to the ability to differentiate the feature of solutions which visited during the search. Operationally, quality becomes a foundation for incentive-based learning, where inducements are provided to reinforce actions that lead to good solutions, and penalties are provided to discourage actions that lead to weak solutions.

d. The influence considers the impact of the choices made during the search on quality and the structure. Recording information about the influence of choices on specific solution elements incorporates an additional level of learning (Glover and Laguna, n.d.).

4. Artificial Immune System (AIS)

Farmer et al. proposed AIS (Farmer, Packard, and Perelson 1986). AISs are algorithms that inspired by the principles of the human immune system with the characteristics of self-organization, learning, memory, adaptation, classification, pattern recognition, and robustness(M. Wang et al. 2017),(Read, Andrews, and Timmis 2012). The immune system in organisms comprises two phases, recognition and response. It includes a variety of specialized cells that distribute and monitor the body to detect and recognize pathogens within the body. This is followed by a specific effector response aimed at eliminating the pathogen. These phases are very complicated, with many details not yet adequately understood. Once pathogen invades the body, the pathogen becomes *antigen* and stimulates the immune response by propagating a mass of cloned *antibodies* to destroy the antigen. If such an antigen is detected for the first time, which is referred to as the *primary response*. While the next time that pathogen is encountered, a faster and often more aggressive response is mounted, and this referred to as the *secondary response*. The immune system offers the ability to learn by its ability to recognize previously unseen pathogens and to remember previously encountered pathogens. The immune system also is capable of recognizing a diverse range of patterns; this is achieved through receptors that perceive antigenic materials in differing contexts. Receptors of the *innate immune system* (Generic defense mechanisms that are nonspecific to particular examples of a pathogen, but act against general classes of a pathogen) very little. While receptors of the *adaptive immune system* (Adapt to previously unseen pathogens based upon exposure to them), such as antibodies and T-cell receptors, are subject to vast diversity. The immune system is very effective at discriminatory, harmful substances (usually viewed as nonself) from the body's own tissues (usually viewed as self), and directing its actions accordingly(Read, Andrews, and Timmis 2012).

Most AISs and their variations were derived from the following four major models or theories, clone selection(Burnet 1959), negative selection (Nossal 1994), Dendritic Cell (Gu, Greensmith, and Aickelin 2013), and artificial immune network(Jerne 1974).

The primary phases in AIS as the following (Zheng, Chen, and Zhang 2010; M. Wang et al. 2017):

- *Initialization*: Candidate antibodies are randomly produced to form the population.
- *Evaluation*: The affinity of these antibodies is evaluated.
- *Cloning*: In each time epoch, any antibody is cloned for several offsprings. The higher the affinity of parent antibody is, the more offsprings are reproduced. Furthermore, the number of clones is nonlinear to the affinity of the parent antibody.
- *Mutation*: (the higher the affinity, the smaller the mutation rate, and vice-versa). These clones, except for the parent one experience the mutation operator. Among these clones, only the one with the highest affinity is selected to remain. Antibodies experience learning mechanisms to accelerate the convergence process. Moreover, a dynamic searching radius update strategy is designed to improve the solution accuracy to maintain the tradeoff between exploration and exploitation
- *Suppression:* the suppression operator is activated if the current average affinity is not significantly different from that of the previous time epoch. Antibodies whose similarity is less than a suppression threshold (except the one with the highest affinity) are suppressed. Then, several randomly generated antibodies are recruited. There are interactions between not only antibodies and antigens but also antibodies themselves, which cause suppression of the antibodies. The higher the affinity is, the better the candidate antibody matches the particular antigen.
- *Repeat* these iterative process until the termination condition is met, such as minimum pattern recognition or classification error.

5. Heart Algorithm (HA)

Heart algorithm proposed by Abdolreza Hatamlou (Hatamlou 2014). HA employs the action of the heart and circulatory system of human beings. This algorithm starts with a random initial population of candidate solutions over the search space and objective function, which is computed for them. The best candidate solution is selected as the heart, and the other candidates form blood molecules. Then, the heart forces other candidates to go away from the heart and start a new search for the optimal solution. The main steps in the heart algorithm are:

Step1*: Initialization.* Generate randomly initial population of candidate solutions over the search space

Step2: *Evaluation.* Evaluate the fitness of all the candidate solutions.

Step3: *Construction.* Choose the best candidate who has the best fitness value as a heart and all other candidates as the blood molecules.

Step4: *Contraction.* Move all the blood molecules toward the heart. The contraction of the heart occurs, and the heart pumps these molecules to outside (search space). This action causes the candidate solutions to efficiently search the problem space.

Step5: *Expansion.* If there is a blood molecule that has reached the same position of the heart, then move it away from the heart. The objective function again calculates the fitness value of each candidate. If there is a molecule that has a better fitness value than the heart, it exchanges its positions, and that molecule becomes the new heart, and the old heart changes to a normal molecule.

Step6: *Check the termination criteria.* If the termination criterion is met, the algorithm stops, otherwise go to Step2.

6. Kidney Algorithm (KA)

Najmeh Sadat Jaddi et al. suggested a new inspired method called kidney algorithm (Jaddi, Alvankarian, and Abdullah 2017). This method inspired by the kidney process in the human body system. Kidneys are the main biological structure of the urine system in the human body. The kidney method was summarized into four steps (Jaddi, Alvankarian, and Abdullah 2017): Filtration, Reabsorption, Secretion, and Excretion.

a. *Filtration.* Moving solutes and water from the blood to the tubule. Particles in water and solute in plasma are represented in the KA as solutions. A continuous flow of fluids into a kidney is represented as generating a new random solution. Filtering useful particles from harmful particles represent separating better solutions from worse solutions.

b. *Reabsorption.* Transportation useful water and solutes from the tubule back into the bloodstream.

c. *Secretion.* Transfering extra and harmful substances from the blood into the tubular.

d. *Excretion.* Passing the waste results of the earlier three procedures into the urine. (Remove worst solutions)

7. Artificial Coronary Circulation System (ACCS)

Ali Kafeh and Mohsen kooshkebaghi proposed a new swarm intelligence optimization technique called Artificial Coronary Circulation System (Kaveh and Kooshkebaghi 2019a). ACCS is an Optimization method inspired by coronary arteries' growth in the human heart. In this optimization algorithm, an initial group of capillaries is considered as population, and the different main arteries are considered as different design variables of the optimization problem. A capillary leader's result is like the "fitness value" of the optimization problem. In this algorithm, any branch of the coronary tree is considered as a new solution. The total cost of the tree at any end is considered as the cost of the object function at any solution. With these values, the Coronary Growth Factor (CGF) is calculated for any solution. This result is similar to the fitness value of the optimization problem. Also, the best solution in the entire population is considered as the stem, and the best solution is the minimum value of the CGF. The process of the ACCS is divided into two main parts, the 'Bifurcation phase' and 'Pruning phase'. The ACCS procedure can be briefly outlined as followed steps after enhancing in EACCS (Kaveh and Kooshkebaghi 2019b):

Step 1: Initialization. The initial positions for CLs are determined randomly in the search space.

Step 2: Defining Coronary Growth Factor (CGF).
In this step, the value of CGF for each CL is evaluated.

Step 3: Storing. Heart memory (HM) is utilized to save several historically best capillary leaders (CL vectors) with their related CGF and the objective function values. Solution vectors that are saved in the HM are added to the population, whereas the same number of current worst CLs is deleted. Finally, CLs are sorted according to their objective function values in increasing order.

Step 4: *Bifurcation and Pruning phase and Jump out* (global search). To improve the exploration capability of the standard ACCS and to prevent premature convergence, a stochastic approach is used in EACCS.

Step 5: Runner and Pruning phase (local search); in this step, occurs Updating CLs, and the new position of each CL. The updated CLs are sorted according to their objective function values.

Step 6: Updating Heart Memory. Keeping the best CLs in each iteration.

Step 7: Terminating condition check. The process of optimization is terminated after the predefined maximum evaluation number. Otherwise, the process is continued by going to Step 2.

III. COMPARISON BETWEEN HUMAN BODY SYSTEM ALGORITHMS

This section covers the comparison between human body system algorithms and their solution methodologies (Table 2). This comparison is based on the following criteria:

a) Continuous vs. Discrete

There are different classes of optimization problems that differ in the definition of the search space (Mavrovouniotis, Li, and Yang 2017). In this paper, two fundamental types of problems are considered as follows:

Discrete optimization problems (All optimization variables are discrete). Discrete optimization problems have a finite search space, e.g., binary that is restricted to the values 0 and 1.

Continuous optimization problems. (All optimization variables are continuous). In continuous optimization problems, each variable value may assume an infinite number of values, e.g., real numbers. Therefore, representing continuous variables requires to impose specific limitation since it is not possible to represent an infinite number of values.

b) Exploration & Exploitation

One way of analyzing algorithms is to look at their exploration and exploitation abilities (global search and local search, respectively. In machine learning algorithms, exploration and exploitation correspond to acquisition and utilization of knowledge, respectively.

Exploration supports diversification, which allows the algorithm to search different regions in the design space and thus increases the probability of finding the true global optimality. Exploration is often achieved by randomization in terms of some predefined probability distributions. A key feature of traditional algorithms is that they can enhance the exploitation ability, but lacks exploration capabilities because they are mainly deterministic and no randomness is used in generating new solutions. In contrast, nature-inspired metaheuristic algorithms use a certain degree of randomness. A reasonable degree of randomness will increase the exploration ability but may reduce the exploitation abilities.

Exploitation: uses local information such as gradients to search local regions more intensively, and such intensification can enhance the rate of convergence. Exploitation can make the population less diverse, and reliable local guidance can even make the population relatively uniform in terms of solution variations.

c) Equations

Equations in algorithms usually represent the iterative evolution of solutions and summarize the method mathematically.

VI. HUMAN BODY SYSTEM- INSPIRED ALGORITHMS APPLICATIONS

Metaheuristic algorithms have been used on a variety of tasks in different areas, including cloud computing, artificial intelligence, internet of things (IoT), security, medical diagnosis, quantum…etc. This section (Table 3) shows some areas and applications that have used HBSIAs as solutions and methods for solving modern problems and technologies.

Table 2 : Comparison between HBSMAs

Algorithm	***Initial problem domain***	***Exploration Mechanism***	***Exploitation Mechanism***	***First solution (global search)***	***Next solutions (local search)***	***Abbreviation of equations***	***Ref.***
ANN	Continuous & Discrete	Training phase	Learning phase	$\boldsymbol{w_{ij}}^{\boldsymbol{t+1}} = \boldsymbol{w_{ij}}^{\boldsymbol{t}} - \alpha \frac{\partial \mathrm{L}}{\partial \boldsymbol{w_{ij}}}$	$\boldsymbol{y_j} = f(\sum_{i=1}^{n} \boldsymbol{w_{ij}} \times \boldsymbol{x_i} + \mathrm{b})$	$w_{ij} . x_i . y_j . b$ are the weights ,input activations , output activations, and bias respectively, α is called the learning rate, L is loss (The gap between the ideal correct scores and the scores computed by the DNN based on its current weights),f is a non-linear function, and n is number of inputs.(Figure 2)	(McCulloch and Pitts 1943; Read, Andrews, and Timmis 2012),(Sze et al. 2017)
GA	Continuous & Discrete	Crossover & mutation operators	Selection operator	$\boldsymbol{S_c(s)} \geq \left[1 - \mathrm{P_c} \frac{\boldsymbol{d(s)}}{\mathrm{l} - 1}\right]$ $\boldsymbol{S_m(s)} = (1 - \mathrm{P_m})^{o(s)}$	$\mathrm{E}\big(\mathrm{N}(\mathrm{s.t} + 1)\big) \geq \frac{\hat{\mathbf{u}}(\mathbf{s.t})}{\bar{\mathbf{F}}(\mathbf{t})} \mathrm{N}(\mathrm{s.t}) \cdot \left[1 - \mathrm{P_c} \frac{\boldsymbol{d(s)}}{\mathrm{l} - 1}\right] \cdot \left[(1 - \mathrm{P_m})^{o(s)}\right]$, $\hat{\boldsymbol{u}}(\boldsymbol{s.t}) = (\sum_{x \in s} \boldsymbol{F(x)}) / \mathrm{N}(\mathrm{s.t})$	$\hat{u}(s.t)$ is the average fitness of instances of s in the population at time t , $S_c(s)$ is the probability of survival under crossover, $S_m(s)$ is the probability that schema s will survive under mutation, $E\big(N(s.t+1)\big)$ is the expected number of instances of s at time t + 1, $N(s.t)$ is number of instances of s at time t, P_c is the probability that single-point crossover will be applied to a string, P_m is the probability of any bit being mutated, d(s) is the defining length of s , L is the length of bit strings in the search space, $o(s)$ is the order of s (i.e., the number of defined bits in s), F(x) is the fitness of string x in the population and $\bar{F}(t)$ is the average fitness of the population at time t.	(1975; Mitchell 1995)
TS	Continuous & Discrete	Intermediate-term memory	Long term memory	***Step*** = ((0.1*$\boldsymbol{w_{best}}$) - (0.2*$\boldsymbol{w_{best}}$) * random) / change)	***Step*** = ((0.1*$\boldsymbol{w_{best}}$) - (0.2*$\boldsymbol{w_{best}}$) * random) * change	Step is the step value, w_{best} the best solution found, random is random number in the range [0, 1], and the change variable is initialized to one, and is increased in one after each intensification phase.	(Glover 1986; Glover and Martí, n.d.)

AIS	Continuous & Discrete	Cloning operator	mutation operator	$\boldsymbol{Aff^*(i)} = (\boldsymbol{Aff}\big(\boldsymbol{Ab_i(t)}\big) - \boldsymbol{Aff_{min}}) / (\boldsymbol{Aff_{max}} - \boldsymbol{Aff_{min}})$
				$\boldsymbol{Nc_i(t)} = round((\boldsymbol{Nc_{max}} - \mathrm{Nc_{min}}) * \boldsymbol{Aff^*}\big(\boldsymbol{Ab_i(t)}\big)^{\boldsymbol{n}} + \mathrm{Nc_{min}})$
					$\boldsymbol{Ab_i(t+1)} = \boldsymbol{Ab_i(t)} + \Delta \boldsymbol{Ab_i(t)}$
					If (Aff $(\boldsymbol{Ab_i(t)}) < \boldsymbol{Aff}(\boldsymbol{Ab_e(t)})$
					$\Delta \boldsymbol{Ab_i(t)}$=rand $* (\boldsymbol{Ab_e(t)} - \boldsymbol{Ab_i(t)})$
					else
					$\Delta \boldsymbol{Ab_i(t)} = \mathrm{rand}\, n * \boldsymbol{\lambda_i}(\mathbf{t})$
						$Ab_i(t)$ is the ith individual (antibody) in the t time epoch, $Aff(\cdot)$ is affinity function to evaluate antibodies. $Aff_{max} . Aff_{min}$ are the maximum and minimum of individuals affinities in the time epoch respectively, $Nc_{max} . Nc_{\mathrm{min}}$are the maximum and minimum number of offsprings, respectively, and n is the power factor of the control function rand is a uniform random variable, rand n is a Gaussian random variable with zero mean and standard deviation1 ,$Ab_e(t)$ is the best antibody in affinity, λ_i(t) is the searching radius of $Ab_i(t)$ in the ith time epoch.	(Farmer, Packard, and Perelson 1986; M. Wang et al. 2017)

KA	Continuous	Reabsorption phase	Filtering phase	$\boldsymbol{S_{i+1} = S_i +}$ rand $\cdot (\boldsymbol{S_{best} - S_i})$	$\boldsymbol{fr} = \alpha \times \frac{\boldsymbol{\sum_{i=1}^{p} f(X_i)}}{\mathrm{p}}$	S represents a solution in the population , $rand$ is a random number between 0 and a given number, S_{best} is the best solution founded by KA, fr is filtration rate, α is constant value between (0,1], $f(X_i)$ is the objective function of solution x at iteration i, and p is the population size.	(Jaddi, Alvankarian, and Abdullah 2017)
HA	Continuous	Expansion phase	Contraction phase	$\boldsymbol{X_i}^{\boldsymbol{new}} = \boldsymbol{X_i}^{\boldsymbol{current}} +$ rand $\cdot (\boldsymbol{X_{heart}} - \boldsymbol{X_i}^{\boldsymbol{current}})$	$\boldsymbol{X_i}^{\boldsymbol{new}} = \boldsymbol{X_{heart}} +$ rand $\cdot$ p $\cdot \boldsymbol{X_{heart}}$	$X_i^{current}.X_i^{new}$ are the current and new position the ith molecule in the search space, respectively, X_{heart} is the position of the current heart in the search space. $rand$ is a random number between [0,1]. p is a decreasing function of time, which controls the power of the heart.	(Hatamlou 2014)
ACCS	Continuous	Bifurcation phase	Pruning phase	$\boldsymbol{X_0}^{\boldsymbol{i}} = \boldsymbol{X_{min}} +$ rand $\cdot (\boldsymbol{X_{max}} - \boldsymbol{X_{min}})$	$\boldsymbol{X_{i.j}}^{\boldsymbol{t+1}} = \boldsymbol{X_{i.j}}^{\boldsymbol{t}} + \alpha \cdot$ rand $\cdot (\boldsymbol{X_{b.j}}^{\boldsymbol{t}} - \boldsymbol{X_{w.j}}^{\boldsymbol{t}})$	$X_{i.j}{}^{0}$ is the initial solution vector of the ith CL, X_{min} and X_{max} are the bounds of design variables; $rand$ is a random vector with each component being in the interval [0,1]; $X_{i.j}{}^{t+1}.X_{i.j}{}^{t}$ are the new and old position value of the jth variable for the ith CL, α is angiogenesis index and it is a decreasing function of time, which controls the pumping power of the heart, of (0,1); and $X_{b.j}{}^{t}.X_{w.j}{}^{t}$) are the best and the worst CLs of the last (old) population which refer to the best arteries leader and the worst arteries leader.	(Kaveh and Kooshkebaghi 2019a; 2019b)

V. CONCLUSION

Several methods have been proposed to solve and optimize difficult computation optimization problems. Algorithms inspired from the natural behavior yields special attention recently for its performance. Because These natural techniques of computing inspired by precise and optimal systems in real life; therefore, these methods provide several ways for solving real-world problems efficiently. This paper covers the human body system inspired algorithms in various fields for accomplishing various tasks. Also, it compares these algorithms in their solution methodologies, and it shows some applications to improve HBSIAs success in practice. This research aims to serve as a source of inspiration for new researchers to continue and build in this field. As a result, it helps to develop new methods that may be inspired by any organ or system in the human body system to solve a wide range of tough optimization more efficiently and more accurately.

VI. ACKNOWLEDGMENTS

Many thanks for academic staff of Faculty of Computer and Information Technology (FCIT) for providing research support.

Table 3: HBSIAs Applications

Algorithms / **Applications**	**ANN**	**GA**	**TS**	**AIS**	**KS**	**Heart**	**ACCS**	**Reference**
Optimization	√	√	√	√	√	√	√	(Hatata, Osman, and Aladl 2017)(Sivaram et al. 2019)
Automatic Programming		√	√	√				(S. Li et al. 2018; L.-C. Yang et al. 2006; Banerjee 2017; Díaz, Tuya, and Blanco 2003)
Machine learning	√	√	√	√				(Watkins, Timmis, and Boggess 2004; Roy et al. 2017; Ali, Masmoudi, and Dhouib 2016; Wei and Ni 2018)
Cloud computing	√	√	√	√				(Dasgupta et al. 2013; Hormozi et al. 2013; Al-Douri et al. 2012; Shu, Wang, and Wang 2014; R.-L. Zhang et al. 2014; Elzamly et al. 2017; Kirci 2016)
Routing			√	√	√			(Kirci 2016; Keko, Skok, and Skrlec 2003; Farahani, Farzin, and Karami 2018)
IoT	√	√		√				(C. M. Liu et al. 2012; C. Liu, Zhang, and Zhang 2013; Y. Wang et al. 2018; Hodo et al. 2016)
Prediction	√	√	√	√	√			(Chatterjee et al. 2018; Catal, Diri, and Ozumut 2007; Singh and Sengupta 2007; Chatterjee et al. 2017; Vui et al. 2013; Lin and Zhang 2014; Han, Li, and Liu 2011; Jaddi and Abdullah 2019)
Classification/ Clustering		√	√	√	√	√		(Hatamlou 2014; Jaddi and Abdullah 2019; Al-Sultan 1995; Casillas, De Lena, and Martínez 2003; Anaya, Tibaduiza, and Pozo 2017)
Image processing	√	√	√					(Daniel 2018; Jiang and Yang 2002; Nikbakht Aali and Bagherzadeh 2019; Hatou, Pamungkas, and Morimoto 2011)
Medicine	√	√		√				(Latifoğlu et al. 2007; Polat, Şahan, and Güneş 2006; Chikh, Saidi, and Settouti 2012; Rashid and Abdullah 2018)
Search engine		√						(Nick and Themis 2001; Thada and Jaglan 2013)
Scheduling	√	√	√	√				(Mori, Tsukiyama, and Fukuda 1998; N. Zhang et al. 2018; Agarwal, Tiwari, and Mukherjee 2007; Bagheri et al. 2010; Coello, Rivera, and Cortes 2003; Bai and Shahidehpour 1996; Ferland et al. 2001; Lü and Hao 2010; Sim, Yeo, and Lee 1994)
Planning	√	√	√	√	√			(P. K. Das et al. 2012; Hu, Xie, and Xu 2007; Y.-N. Wang, Hsu, and Lin 2005; Masehian and Amin-Naseri 2008; Ismail, Sheta, and Al-Weshah 2008; Shibata and Fukuda 1993; Pradipta Kumar Das 2019)
Diabetic Mellitus Diagnosing	√	√		√				(Rashid and Abdullah 2018; Chikh, Saidi, and Settouti 2012; Peng et al. 2008; Polat, Şahan, and Güneş 2006)
Feature selection		√	√	√	√			(Taqi and Ali 2017; X.-Y. Liu et al. 2018; Castro and Von Zuben 2010; Kuo et al. 2013; H. Zhang and Sun 2002)
Cryptology	√	√	√	√				(Ibrahim and Maarof 2005; Garg 2009; Crainicu and Enachescu 2011)
Data mining		√	√	√				(Timmis and Knight 2002; Djenouri, Drias, and Chemchem 2013; S. Das and Saha 2009; Ghosh and Nath 2004)
Intrusion detection	√	√	√	√				(Kim et al. 2007; Roy et al. 2017; Jian-guang, Ran, and Zhi-Yong 2011; Z. Yang et al. 2008; Khan 2011; Xia et al. 2005)
Quantum		√	√	√				(Abdullah, Tsimenidis, and Johnston 2018; Zhao et al. 2018; Y. Li and Jiao 2005; Gao et al. 2014)

VII. REFERENCES

1. Abdullah, Zaid, Charalampos C. Tsimenidis, and Martin Johnston. 2018. “Quantum-Inspired Tabu Search Algorithm for Antenna Selection in Massive MIMO Systems.” In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6. IEEE.
2. Agarwal, Rina, Manoj K. Tiwari, and Sanat K. Mukherjee. 2007. “Artificial Immune System Based Approach for Solving Resource Constraint Project Scheduling Problem.” *The International Journal of Advanced Manufacturing Technology* 34 (5–6): 584–593.
3. Al-Douri, Yamur K., Ramin Karim, Aditya Parida, and Uday Kumar. 2012. “Model—Based Security System for Data Acquisition in e-Maintenance Using Artificial Immune System and Cloud Computing.” *International Journal of Condition Monitoring and Diagnostic Engineering Management* 15 (4): 26.
4. Ali, Bilel Ben, Youssef Masmoudi, and Souhail Dhouib. 2016. “DTW-Global Constraint Learning Using Tabu Search Algorithm.” *Procedia Computer Science* 82: 12–19.
5. Al-Sultan, Khaled S. 1995. “A Tabu Search Approach to the Clustering Problem.” *Pattern Recognition* 28 (9): 1443–1451.
6. Anaya, Maribel, Diego Alexander Tibaduiza, and Francesc Pozo. 2017. “Detection and Classification of Structural Changes Using Artificial Immune Systems and Fuzzy Clustering.” *International Journal of Bio-Inspired Computation* 9 (1): 35–52.
7. Bagheri, A., Mostafa Zandieh, Iraj Mahdavi, and Mehdi Yazdani. 2010. “An Artificial Immune Algorithm for the Flexible Job-Shop Scheduling Problem.” *Future Generation Computer Systems* 26 (4): 533–541.
8. Bai, Xaiomin, and S. M. Shahidehpour. 1996. “Hydro-Thermal, Scheduling by Tabu Search and Decomposition Method.” *IEEE Transactions on Power Systems* 11 (2): 968–974.
9. Banerjee, Soumya. 2017. “An Artificial Immune System Approach to Automated Program Verification: Towards a Theory of Undecidability in Biological Computing.” PeerJ Preprints.
10. Burnet, Sir Frank Macfarlane. 1959. “The Clonal Selection Theory of Acquired Immunity.”
11. Casillas, Arantza, MT González De Lena, and R. Martínez. 2003. “Document Clustering into an Unknown Number of Clusters Using a Genetic Algorithm.” In *International Conference on Text, Speech and Dialogue*, 43–49. Springer.
12. Castro, Pablo AD, and Fernando J. Von Zuben. 2010. “Multi-Objective Feature Selection Using a Bayesian Artificial Immune System.” *International Journal of Intelligent Computing and Cybernetics*.
13. Catal, Cagatay, Banu Diri, and Bulent Ozumut. 2007. “An Artificial Immune System Approach for Fault Prediction in Object-Oriented Software.” In *2nd International Conference on Dependability of Computer Systems (DepCoS-RELCOMEX’07)*, 238–245. IEEE.
14. Chatterjee, Sankhadeep, Nilanjan Dey, Amira S. Ashour, and Cornelia Victoria Anghel Drugarin. 2018. “Electrical Energy Output Prediction Using Cuckoo Search Based Artificial Neural Network.” In *Smart Trends in Systems, Security and Sustainability*, 277–285. Springer.
15. Chatterjee, Sankhadeep, Sarbartha Sarkar, Nilanjan Dey, Soumya Sen, Takaaki Goto, and Narayan C. Debnath. 2017. “Water Quality Prediction: Multi Objective Genetic Algorithm Coupled Artificial Neural Network Based Approach.” In *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*, 963–968. IEEE.
16. Chikh, Mohamed Amine, Meryem Saidi, and Nesma Settouti. 2012. “Diagnosis of Diabetes Diseases Using an Artificial Immune Recognition System2 (AIRS2) with Fuzzy K-Nearest Neighbor.” *Journal of Medical Systems* 36 (5): 2721–2729.
17. Coello, Carlos A. Coello, Daniel Cortes Rivera, and Nareli Cruz Cortes. 2003. “Use of an Artificial Immune System for Job Shop Scheduling.” In *International Conference on Artificial Immune Systems*, 1–10. Springer.
18. Colorni, Alberto, Marco Dorigo, and Vittorio Maniezzo. 1992. “Distributed Optimization by Ant Colonies.” In *Proceedings of the First European Conference on Artificial Life*, 142:134–142. Cambridge, MA.
19. Crainicu, Bogdan, and Calin Enachescu. 2011. “A Metaheuristic Tabu Search Approach for Internal State Reconstruction of RC4 Stream Cipher.” In *2011 RoEduNet International Conference 10th Edition: Networking in Education and Research*, 1–4. IEEE.
20. Daniel, Ebenezer. 2018. “Optimum Wavelet-Based Homomorphic Medical Image Fusion Using Hybrid Genetic–Grey Wolf Optimization Algorithm.” *IEEE Sensors Journal* 18 (16): 6804–6811.
21. Das, P. K., S. K. Pradhan, S. N. Patro, and B. K. Balabantaray. 2012. “Artificial Immune System Based Path Planning of Mobile Robot.” In *Soft Computing Techniques in Vision Science*, 195–207. Springer.
22. Das, Pradipta Kumar. 2019. “Hybridization of Kidney-Inspired and Sine–Cosine Algorithm for Multi-Robot Path Planning.” *Arabian Journal for Science and Engineering*, 1–18.

23. Das, Sufal, and Banani Saha. 2009. “Data Quality Mining Using Genetic Algorithm.” *International Journal of Computer Science and Security* 3 (2): 105–112.
24. Dasgupta, Kousik, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mandal, and Santanu Dam. 2013. “A Genetic Algorithm (Ga) Based Load Balancing Strategy for Cloud Computing.” *Procedia Technology* 10: 340–347.
25. Díaz, Eugenia, Javier Tuya, and Raquel Blanco. 2003. “Automated Software Testing Using a Metaheuristic Technique Based on Tabu Search.” In *18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings.*, 310–313. IEEE.
26. Djenouri, Youcef, Habiba Drias, and Amine Chemchem. 2013. “A Hybrid Bees Swarm Optimization and Tabu Search Algorithm for Association Rule Mining.” In *2013 World Congress on Nature and Biologically Inspired Computing*, 120–125. IEEE.
27. Elzamly, Abdelrafe, Burairah Hussin, Samy S. Abu-Naser, Tadahiro Shibutani, and Mohamed Doheir. 2017. “Predicting Critical Cloud Computing Security Issues Using Artificial Neural Network (ANNs) Algorithms in Banking Organizations.”
28. Farahani, Nazanin Node, Saeed Farzin, and Hojat Karami. 2018. “Flood Routing by Kidney Algorithm and Muskingum Model.” *Natural Hazards*, 1–19.
29. Farmer, J. Doyne, Norman H. Packard, and Alan S. Perelson. 1986. *North-Holland, Amsterdam THE IMMUNE SYSTEM, ADAPTATION, AND MACHINE LEARNING*.
30. Ferland, Jacques A., Soumia Ichoua, Alain Lavoie, and Eric Gagné. 2001. “Scheduling Using Tabu Search Methods with Intensification and Diversification.” *Computers & Operations Research* 28 (11): 1075–1092.
31. Gao, Jiaquan, Guixia He, Ronghua Liang, and Zhilin Feng. 2014. “A Quantum-Inspired Artificial Immune System for the Multiobjective 0–1 Knapsack Problem.” *Applied Mathematics and Computation* 230: 120–137.
32. Garg, Poonam. 2009. “GENETIC ALGORITHMS, TABU SEARCH AND SIMULATED ANNEALING: A COMPARISON BETWEEN THREE APPROACHES FOR THE CRYPTANALYSIS OF TRANSPOSITION CIPHER.” *Journal of Theoretical & Applied Information Technology* 5 (4).
33. Geem, Zong Woo, Joong Hoon Kim, and Gobichettipalayam Vasudevan Loganathan. 2001. “A New Heuristic Optimization Algorithm: Harmony Search.” *Simulation* 76 (2): 60–68.
34. Ghosh, Ashish, and Bhabesh Nath. 2004. “Multi-Objective Rule Mining Using Genetic Algorithms.” *Information Sciences* 163 (1–3): 123–133.
35. Glover, Fred. 1986. “Future Paths for Integer Programming and Links to Artificial Intelligence.” *Computers & Operations Research* 13 (5): 533–549.
36. Glover, Fred. 1989. “Tabu Search—Part I.” *ORSA Journal on Computing* 1 (3): 190–206.
37. Glover, Fred, and Manuel Laguna. n.d. “TABU SEARCH.”
38. Glover, Fred, and Rafael Martí. n.d. “TABU SEARCH.” In .
39. Gu, Feng, Julie Greensmith, and Uwe Aickelin. 2013. “Theoretical Formulation and Analysis of the Deterministic Dendritic Cell Algorithm.” *Biosystems* 111 (2): 127–135.
40. Han, Shuang, Jinshan Li, and Yongqian Liu. 2011. “Tabu Search Algorithm Optimized ANN Model for Wind Power Prediction with NWP.” *Energy Procedia* 12: 733–740.
41. Hatamlou, Abdolreza. 2014. “Heart: A Novel Optimization Algorithm for Cluster Analysis.” *Progress in Artificial Intelligence* 2 (2–3): 167–73. https://doi.org/10.1007/s13748-014-0046-5.
42. Hatata, Ahmed Youssef, Mohamed Galal Osman, and Mohamed M. Aladl. 2017. “A Review of the Clonal Selection Algorithm as an Optimization Method.” *Leonardo Journal of Sciences* 16 (30): 1–14.
43. Hatou, Kenji, Agung Putra Pamungkas, and Tetsuo Morimoto. 2011. “Image Processing by Artificial Neural Networks for Stress Diagnosis of Tomato.” *IFAC Proceedings Volumes* 44 (1): 1768–1772.
44. Hodo, Elike, Xavier Bellekens, Andrew Hamilton, Pierre-Louis Dubouilh, Ephraim Iorkyase, Christos Tachtatzis, and Robert Atkinson. 2016. “Threat Analysis of IoT Networks Using Artificial Neural Network Intrusion Detection System.” In *2016 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6. IEEE.
45. Holland, J. H.,. 1975. “Adaptation in Natural and Artificial Systems.” *University of Michigan Press, Ann Arbor, MI*.
46. Hormozi, Elham, Mohammad Kazem Akbari, Morteza Sargolzaei Javan, and Hadi Hormozi. 2013. “Performance Evaluation of a Fraud Detection System Based Artificial Immune System on the Cloud.” In *2013 8th International Conference on Computer Science & Education*, 819–823. IEEE.
47. Hu, Xuanzi, Cunxi Xie, and Qingui Xu. 2007. “Robot Path Planning Based on Artificial Immune Network.” In *2007 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1053–1057. IEEE.
48. Hussain, Kashif, Mohd Najib Mohd Salleh, Shi Cheng, and Yuhui Shi. 2019. “Metaheuristic

Research: A Comprehensive Survey.” *Artificial Intelligence Review* 52 (4): 2191–2233.
49. Ibrahim, Subariah, and Mohd Aizaini Maarof. 2005. “A Review on Biological Inspired Computation in Cryptology.” *Jurnal Teknologi Maklumat* 17 (1): 90–98.
50. Ismail, A., Alaa Sheta, and Mohammed Al-Weshah. 2008. “A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment.” *Journal of Computer Science* 4 (4): 341–344.
51. Jaddi, Najmeh Sadat, and Salwani Abdullah. 2019. “Kidney-Inspired Algorithm with Reduced Functionality Treatment for Classification and Time Series Prediction.” *PloS One* 14 (1): e0208308.
52. Jaddi, Najmeh Sadat, Jafar Alvankarian, and Salwani Abdullah. 2017. “Kidney-Inspired Algorithm for Optimization Problems.” *Communications in Nonlinear Science and Numerical Simulation* 42 (January): 358–69. https://doi.org/10.1016/j.cnsns.2016.06.006.
53. Jerne, Niels K. 1974. “Towards a Network Theory of the Immune System.” *Ann. Immunol.* 125: 373–389.
54. Jiang, Tianzi, and Faguo Yang. 2002. “An Evolutionary Tabu Search for Cell Image Segmentation.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 32 (5): 675–678.
55. Jian-guang, Wu, Tao Ran, and Li Zhi-Yong. 2011. “An Improving Tabu Search Algorithm for Intrusion Detection.” In *2011 Third International Conference on Measuring Technology and Mechatronics Automation*, 1:435–439. IEEE.
56. Karaboga, Dervis. 2005. “An Idea Based on Honey Bee Swarm for Numerical Optimization.” Technical report-tr06, Erciyes university, engineering faculty, computer ….
57. Karaboga, Dervis, and Bahriye Basturk. 2007. “Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems.” In *International Fuzzy Systems Association World Congress*, 789–798. Springer.
58. Kaveh, Ali, and Mohsen Kooshkebaghi. 2019a. “Artificial Coronary Circulation System; A New Bio-Inspired Metaheuristic Algorithm.” *Scientia Iranica*.
59. ———. 2019b. “Enhanced Artificial Coronary Circulation System Algorithm for Truss Optimization with Multiple Natural Frequency Constraints.”
60. Keko, Hrvoje, Minea Skok, and Davor Skrlec. 2003. “Artificial Immune Systems in Solving Routing Problems.” In *The IEEE Region 8 EUROCON 2003. Computer as a Tool.*, 1:62–66. IEEE.
61. Kennedy, J., and R. Eberhart. 1995. “Particle Swarm Optimization (PSO).” In *Proc. IEEE International Conference on Neural Networks, Perth, Australia*, 1942–1948.
62. Khan, M. Sadiq Ali. 2011. “Rule Based Network Intrusion Detection Using Genetic Algorithm.” *International Journal of Computer Applications* 18 (8): 26–29.
63. Kim, Jungwon, Peter J. Bentley, Uwe Aickelin, Julie Greensmith, Gianni Tedesco, and Jamie Twycross. 2007. “Immune System Approaches to Intrusion Detection–a Review.” *Natural Computing* 6 (4): 413–466.
64. Kirci, Pinar. 2016. “On the Performance of Tabu Search Algorithm for the Vehicle Routing Problem with Time Windows.” In *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 351–354. IEEE.
65. Kirkpatrick, Scott, C. Daniel Gelatt, and Mario P. Vecchi. 1983. “Optimization by Simulated Annealing.” *Science* 220 (4598): 671–680.
66. Kuo, R. J., M. C. Shieh, J. W. Zhang, and K. Y. Chen. 2013. “The Application of an Artificial Immune System-Based Back-Propagation Neural Network with Feature Selection to an RFID Positioning System.” *Robotics and Computer-Integrated Manufacturing* 29 (6): 431–438.
67. Latifoğlu, Fatma, Halife Kodaz, Sadık Kara, and Salih Güneş. 2007. “Medical Application of Artificial Immune Recognition System (AIRS): Diagnosis of Atherosclerosis from Carotid Artery Doppler Signals.” *Computers in Biology and Medicine* 37 (8): 1092–1099.
68. Li, Siguang, Zhengwen Huang, Liangxiu Han, and Changjun Jiang. 2018. “A Genetic Algorithm Enhanced Automatic Data Flow Management Solution for Facilitating Data Intensive Applications in the Cloud.” *Concurrency and Computation: Practice and Experience* 30 (23): e4844.
69. Li, Yangyang, and Licheng Jiao. 2005. “Quantum-Inspired Immune Clonal Algorithm.” In *International Conference on Artificial Immune Systems*, 304–317. Springer.
70. Lin, Xiaoli, and Xiaolong Zhang. 2014. “Protein Structure Prediction with Local Adjust Tabu Search Algorithm.” In *BMC Bioinformatics*, 15:S1. Springer.
71. Liu, Cai Ming, Si Yu Chen, Yan Zhang, Run Chen, and Kui Liang Guo. 2012. “An IoT Anomaly Detection Model Based on Artificial Immunity.” In *Advanced Materials Research*, 424:625–628. Trans Tech Publ.
72. Liu, Caiming, Yan Zhang, and Huaqiang Zhang. 2013. “A Novel Approach to IoT Security Based on Immunology.” In *2013 Ninth International Conference on Computational Intelligence and Security*, 771–775. IEEE.

73. Liu, Xiao-Ying, Yong Liang, Sai Wang, Zi-Yi Yang, and Han-Shuo Ye. 2018. "A Hybrid Genetic Algorithm with Wrapper-Embedded Approaches for Feature Selection." *IEEE Access* 6: 22863–22874.
74. Lü, Zhipeng, and Jin-Kao Hao. 2010. "Adaptive Tabu Search for Course Timetabling." *European Journal of Operational Research* 200 (1): 235–244.
75. Masehian, Ellips, and Mohammad Reza Amin-Naseri. 2008. "Sensor-Based Robot Motion Planning-a Tabu Search Approach." *IEEE Robotics & Automation Magazine* 15 (2): 48–57.
76. Mavrovouniotis, Michalis, Changhe Li, and Shengxiang Yang. 2017. "A Survey of Swarm Intelligence for Dynamic Optimization: Algorithms and Applications." *Swarm and Evolutionary Computation* 33: 1–17.
77. McCulloch, Warren S., and Walter Pitts. 1943. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *The Bulletin of Mathematical Biophysics* 5 (4): 115–33. https://doi.org/10.1007/BF02478259.
78. Mitchell, Melanie. 1995. "Genetic Algorithms: An Overview." In .
79. Mladenović, Nenad, and Pierre Hansen. 1997. "Variable Neighborhood Search." *Computers & Operations Research* 24 (11): 1097–1100.
80. Mori, Kazuyuki, Makoto Tsukiyama, and Toyoo Fukuda. 1998. "Adaptive Scheduling System Inspired by Immune System." In *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218)*, 4:3833–3837. IEEE.
81. Nick, Zacharis Z., and Panayiotopoulos Themis. 2001. "Web Search Using a Genetic Algorithm." *IEEE Internet Computing* 5 (2): 18–26.
82. Nikbakht Aali, Sahar, and Nader Bagherzadeh. 2019. "Divisible Load Scheduling of Image Processing Applications on the Heterogeneous Star and Tree Networks Using a New Genetic Algorithm." *Concurrency and Computation: Practice and Experience*, e5498.
83. Nossal, G. J. V. 1994. "Negative Selection of Lymphocytes." *Cell* 76 (2): 229–239.
84. Peng, Lingxi, Tao Li, Xiaojie Liu, Caiming Liu, Jinquan Zeng, and Jian Zhang. 2008. "An Artificial Immune Network Based Algorithm for Diabetes Diagnosis." *Protein and Peptide Letters* 15 (5): 484–487.
85. Polat, Kemal, Seral Şahan, and Salih Güneş. 2006. "A New Method to Medical Diagnosis: Artificial Immune Recognition System (AIRS) with Fuzzy Weighted Pre-Processing and Application to ECG Arrhythmia." *Expert Systems with Applications* 31 (2): 264–269.
86. Rashid, Tarik, and Saman Abdullah. 2018. "A Hybrid of Artificial Bee Colony, Genetic Algorithm, and Neural Network for Diabetic Mellitus Diagnosing." *ARO-The Scientific Journal of Koya University* 6 (1): 55–64.
87. Read, Mark, Paul S. Andrews, and Jon Timmis. 2012. "An Introduction to Artificial Immune Systems." In *Handbook of Natural Computing*, edited by Grzegorz Rozenberg, Thomas Bäck, and Joost N. Kok, 1575–97. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-92910-9_47.
88. Roy, Sanjiban Sekhar, Abhinav Mallik, Rishab Gulati, Mohammad S. Obaidat, and P. Venkata Krishna. 2017. "A Deep Learning Based Artificial Neural Network Approach for Intrusion Detection." In *International Conference on Mathematics and Computing*, 44–53. Springer.
89. Shibata, Takanori, and Toshio Fukuda. 1993. "Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic." In *Proceedings of 8th IEEE International Symposium on Intelligent Control*, 565–570. IEEE.
90. Shu, Wanneng, Wei Wang, and Yunji Wang. 2014. "A Novel Energy-Efficient Resource Allocation Algorithm Based on Immune Clonal Optimization for Green Cloud Computing." *EURASIP Journal on Wireless Communications and Networking* 2014 (1): 64.
91. Sim, S. K., K. T. Yeo, and W. H. Lee. 1994. "An Expert Neural Network System for Dynamic Job Shop Scheduling." *THE INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH* 32 (8): 1759–1773.
92. Singh, Rohit, and Raghu Nandan Sengupta. 2007. "Bankruptcy Prediction Using Artificial Immune Systems." In *International Conference on Artificial Immune Systems*, 131–141. Springer.
93. Sivaram, M., K. Batri, Mohammed Amin Salih, and V. Porkodi. 2019. "Exploiting the Local Optima in Genetic Algorithm Using Tabu Search." *Indian Journal of Science and Technology* 12 (1): 9.
94. Sulaiman, Mohd Herwan, Hassan Ibrahim, Hamdan Daniyal, and Mohd Rusllim Mohamed. 2014. "A New Swarm Intelligence Approach for Optimal Chiller Loading for Energy Conservation." *Procedia-Social and Behavioral Sciences* 129: 483–488.
95. Sze, Vivienne, Yu-Hsin Chen, Tien-Ju Yang, and Joel S. Emer. 2017. "Efficient Processing of Deep Neural Networks: A Tutorial and Survey." *Proceedings of the IEEE* 105 (12): 2295–2329.
96. Tan, Ying, and Yuanchun Zhu. 2010. "Fireworks Algorithm for Optimization." In *International Conference in Swarm Intelligence*, 355–364. Springer.
97. Taqi, Mustafa Kadhim, and Rosmah Ali. 2017. "OBKA-FS: AN OPPOSITIONAL-BASED BINARY KIDNEYINSPIRED SEARCH ALGORITHM FOR FEATURE SELECTION."

Journal of Theoretical & Applied Information Technology 95 (1).

98. Thada, Vikas, and Vivek Jaglan. 2013. "Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web Retrieved Documents Using Genetic Algorithm." *International Journal of Innovations in Engineering and Technology* 2 (4): 202–205.
99. Timmis, Jonathan, and Thomas Knight. 2002. "Artificial Immune Systems: Using the Immune System as Inspiration for Data Mining." In *Data Mining: A Heuristic Approach*, 209–230. IGI Global.
100. Vui, Chang Sim, Gan Kim Soon, Chin Kim On, Rayner Alfred, and Patricia Anthony. 2013. "A Review of Stock Market Prediction with Artificial Neural Network (ANN)." In *2013 IEEE International Conference on Control System, Computing and Engineering*, 477–482. IEEE.
101. Wang, Mingan, Shuo Feng, Chunhui He, Zhonghua Li, and Yu Xue. 2017. "An Artificial Immune System Algorithm with Social Learning and Its Application in Industrial PID Controller Design," 13.
102. Wang, Yen-Nien, Hao-Hsuan Hsu, and Chun-Cheng Lin. 2005. "Artificial Immune Algorithm Based Obstacle Avoiding Path Planning of Mobile Robots." In *International Conference on Natural Computation*, 859–862. Springer.
103. Wang, Yingchen, Xiaoxiao Geng, Fan Zhang, and Junhu Ruan. 2018. "An Immune Genetic Algorithm for Multi-Echelon Inventory Cost Control of IOT Based Supply Chains." *IEEE Access* 6: 8547–8555.
104. Watkins, Andrew, Jon Timmis, and Lois Boggess. 2004. "Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm." *Genetic Programming and Evolvable Machines* 5 (3): 291–317.
105. Wei, Changyun, and Fusheng Ni. 2018. "Tabu Temporal Difference Learning for Robot Path Planning in Uncertain Environments." In *Annual Conference Towards Autonomous Robotic Systems*, 123–134. Springer.
106. Xia, Tao, Guangzhi Qu, Salim Hariri, and Mazin Yousif. 2005. "An Efficient Network Intrusion Detection Method Based on Information Theory and Genetic Algorithm." In *PCCC 2005. 24th IEEE International Performance, Computing, and Communications Conference, 2005.*, 11–17. IEEE.
107. Yang, Li-Cai, Lei Jia, Qing-Jie Kong, and Wen-Xing Zhu. 2006. "Method of Automatic Programming Traffic Intervals Based on Artificial Immune Algorithm." *Kongzhi Lilun Yu Yingyong/ Control Theory & Applications* 23 (2): 193–198.
108. Yang, Xin-She. 2008. "Firefly Algorithm." *Nature-Inspired Metaheuristic Algorithms* 20: 79–90.
109. Yang, Xin-She. 2010a. "A New Metaheuristic Bat-Inspired Algorithm." In *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 65–74. Springer.
110. Yang, Xin-She. 2010b. *Nature-Inspired Metaheuristic Algorithms - Second Edition*. University of Cambridge ,Unit Kingdom.
111. Yang, Xin-She. 2018. *Nature-Inspired Algorithms and Applied Optimization*. Vol. 744. Studies in Computational Intelligence. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-67669-2.
112. Yang, Xin-She, Zhihua Cui, Renbin Xiao, Amir Hossein Gandomi, and Mehmet Karamanoglu. 2013. *Swarm Intelligence and Bio-Inspired Computation: Theory and Applications*. Newnes.
113. Yang, Xin-She, and Suash Deb. 2009. "Cuckoo Search via Lévy Flights." In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, 210–214. IEEE.
114. Yang, Zhongxue, Adem Karahoca, Ning Yang, and Nizamettin Aydin. 2008. "Network Intrusion Detection by Using Cellular Neural Network with Tabu Search." In *2008 Bio-Inspired, Learning and Intelligent Systems for Security*, 64–68. IEEE.
115. Zhang, Hongbin, and Guangyu Sun. 2002. "Feature Selection Using Tabu Search Method." *Pattern Recognition* 35 (3): 701–711.
116. Zhang, Nan, Xiaolong Yang, Min Zhang, Yan Sun, and Keping Long. 2018. "A Genetic Algorithm-Based Task Scheduling for Cloud Resource Crowd-Funding Model." *International Journal of Communication Systems* 31 (1): e3394.
117. Zhang, Ren-Long, Mi-Yuan Shan, Xiao-Hong Liu, and Li-Hong Zhang. 2014. "A Novel Fuzzy Hybrid Quantum Artificial Immune Clustering Algorithm Based on Cloud Model." *Engineering Applications of Artificial Intelligence* 35: 1–13.
118. Zhao, Wenju, Shuanglin Guo, Yun Zhou, and Jian Zhang. 2018. "A Quantum-Inspired Genetic Algorithm-Based Optimization Method for Mobile Impact Test Data Integration." *Computer-Aided Civil and Infrastructure Engineering* 33 (5): 411–422.
119. Zheng, Jieqiong, Yunfang Chen, and Wei Zhang. 2010. "A Survey of Artificial Immune Applications." *Artificial Intelligence Review* 34 (1): 19–34.

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

ABOUT IJEAST

International Journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high-quality research papers in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

FOCUS AREAS

- Engineering
- Applied Science
- Technology
- Innovation & Development
- Interdisciplinary Studies

PEER REVIEWED

All submissions are rigorously peer reviewed to ensure quality.

OPEN ACCESS

Free and unrestricted access to research for all.

GLOBAL REACH

Connecting researchers and professionals worldwide.

TIMELY PUBLICATION

We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

✉ editor@ijeast.com
www.ijeast.com
India

2455-2143