ISSN 2278-3091
Volume 9, No.1.4, 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse5691.42020.pdf
https://doi.org/10.30534/ijatcse/2020/5691.42020

Review of The Meta-Heuristic Algorithms for Fuzzy Modeling in The Classification Problem

Nur Azieta Mohamad Aseri[1], Mohd Arfian Ismail[2], Abdul Sahli Fakharudin[3], Ashraf Osman Ibrahim[4,5]

[1] Faculty of Computing, Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pahang, Malaysia, azietanur@gmail.com

[2] Faculty of Computing, Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pahang, Malaysia, arfian@ump.edu.my

[3] Faculty of Computing, Faculty of Computing, College of Computing and Applied Sciences, Universiti Malaysia Pahang, Pahang, Malaysia, sahli@ump.edu.my

[4] Faculty of Computer Science and Information Technology, Alzaiem Alazhari University, 13311, Khartoum North, Sudan, ashrafosman2@gmail.com

[5] Arab Open University, Khartoum, Sudan, ashrafosman2@gmail.com

ABSTRACT

Classification is an important tool in today's world, where data is used to make all kinds of decisions in the government, economic, and medicine sectors. Classification is a tool that helps to make sense of the data and find patterns. The increasing size of data has led to more automated approaches as data sets have grown in size and complexity. Fuzzy system is seen as a good approach that can deal with this issue. The fuzzy system works based on implementing fuzzy logic and approximate reasoning. Fuzzy system works well on non-complex issues, but when applied to complex issues such as engineering or medical problems, the construction of the fuzzy system becomes complicated. Due to this, a method is needed where an automatic process to identify the fuzzy parameters is accomplished. There are various algorithms that can be applied to automate the fuzzy modeling. The goal of this study is to review the algorithms that can be applied to automate fuzzy modeling. Based on the literature, Genetic Algorithm, Differential Evolution Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Simulated Annealing, Gravitational Search Algorithm, Imperialist Competitive Algorithm and Teaching-Learning Based Optimization Algorithm were widely used and applied in fuzzy modeling. A comparative analysis between these algorithms was done to find the strengths and drawbacks of whether these algorithms are suitable to apply in this study. This study focused on the algorithm that is suitable to apply in automated fuzzy modeling and the chosen algorithm will be implemented in a future work.

Key words: Classification, Fuzzy Modeling, Meta-Heuristic Algorithm

1. INTRODUCTION

Recently, classification has gained attention from many researchers and become an important research area. Classification is the systematic arrangement of related categories used to group data according to its criteria. The aim of classification is to analyze data and automatically generate a model to predict future behavior. Generally, most of the classification problems contain important information that represents the pattern of the class categories [1]. An early method to analyze data is by manually extracting patterns by humans. However, the increasing size of data has led to more automated approaches as data sets have grown in size and complexity. This has been aided by other discoveries in computer science such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs) and Fuzzy System [1]. In many classification areas, there are several factors that need to consider ensuring the optimum solution can be achieved. ANNs are still frustrating tools in classification areas. They exhibit excellent modeling performance, but do not give a clue about the structure of the models [2]. Meanwhile, SVMs do not deliver a parametric score function and the results produced are normally difficult to understand [3]. In consequence, the fuzzy system is seen as a good system that deals with inaccurate and incomplete issues and highlights the major problems of imprecise, uncertain and unreliable [4].

The fuzzy system works based on implementing fuzzy logic and approximate reasoning and also incorporating expert knowledge in the inference system in order to obtain the desired output from the system input values. Fuzzy parameters are needed while developing the fuzzy system to obtain the desired behavior. This process is called

fuzzy modeling. The fuzzy system works well on the non-complex issues, but when applied to complex issues such as engineering or medical problems, the construction of the fuzzy system becomes complicated [5]. This may happen because of the identification of many fuzzy parameters. Due to the issue, a method is needed to identify the fuzzy parameters where an automatic process of identifying the fuzzy parameters is accomplished.

There are many techniques that can be applied to automate the fuzzy modeling. Current trend shows that using meta-heuristic algorithms is popular and suitable to be used. There are many meta-heuristic algorithms that can be used, and we can categorize them into four main groups, based on their operator and the way they operate which are Evolutionary Algorithm (EA), Swarm Intelligence Algorithm (SIA), Physics Based Algorithm (PBA) and Social Behavior-Based Algorithm (SBBA). The next section discusses these groups and examples of algorithms with their current works.

2. EVOLUTIONARY ALGORITHM

The EA is inspired by biological evolution in nature and the strategies organisms used to interact with each other. Mechanisms used in EA are inspired by biological evolution such as reproduction and recombination. Reproduction attempts to combine elements of existing solutions in order to create a new solution with some of the features of each parent. Recombination performs the process by exchanging genetic information between individuals to produce offspring. The EA starts with the reproduction process by randomly generating the candidate solution. Then, each candidate solution will be evaluated based on its fitness value. After that, the recombination process takes place using a specific operator (based on the algorithm) to produce its offspring. The recombination process continues until termination condition is reached or the maximum number of generations is attained. There are lot of EA algorithms including Genetic Algorithm (GA) [6], Evolutionary Programming [7], Evolutionary Strategy [8], Differential Evolution Algorithm (DE) [9], and Genetic Programming [10]. Among these algorithms, GA and DE are popular choices due to their efficiency and robustness [11]–[13]

2.1. Genetic Algorithm

The GA is a search heuristic which mimics the natural selection process and introduced by John Holland in 1975 [6]. The operation of GA starts with randomly generating a candidate solution. Then, all of the candidate solutions will be evaluated before going into the recombination process. In the recombination process, crossover and mutation will be applied on the candidate solution to produce the next generation of the solution. The recombination process ends when the termination condition is reached, or the maximum number of generations is attained. Figure 1 shows the GA in a pseudocode format.

There are many researchers using GA with fuzzy system in their works. In a work proposed by [14], the authors improved GA and used it in tuning the fuzzy rules and membership function of the fuzzy system. The authors divide the chromosome that represent the fuzzy rules and membership function into multiple sub-chromosomes. They tested their methods on several benchmark datasets and their results performed better compared to existing studies. Another work that implemented the GA into fuzzy system was carried out in [15] where the authors proposed fuzzy genetic algorithm that integrated the GA into the fuzzy system. The use of the GA was to adjust the fuzzy parameters in order to improve the performance of the fuzzy system and make the it robust. Another work that implemented GA into the fuzzy system was performed in [16]. In the work, the authors modeled the rear-end collation control using the fuzzy logic system to control the movement of a car. Then, the authors implemented GA in the system by refining the fuzzy rules to reduce the complexity of the model and maintain the accuracy. The method has been proven by the simulation and the outcome showed that the implementation of GA in the fuzzy system was able to reduce the rear-end collision and increase the efficiency of the system Tables and Figures are presented center, as shown in Table 1 and Figure 1 and cited in the manuscript before appeared.

```
Begin
    Generate randomly an initial population of solutions.
    Calculate the fitness of the initial population.
    Repeat
        Select a pair of parents based on fitness.
        Create two offspring using crossover.
        Apply mutation to each child.
        Evaluate the mutated offspring.
        All the offspring will be the new population, the parents will die.
    Until a stop condition is satisfied.
End.
```

Figure 1: The pseudocode of GA

2.2. Differential Evolution Algorithm

The DE was proposed in 1997 by Stron and Price [9]. Similar with GA, DE is a population-based algorithm that works by reproduction and recombination processes. Unlike GA, DE relies

on mutation operator to produce the offspring and to control the exploration. Meanwhile, the crossover is used to control exploitation. Originally, DE uses the real-coded representation, compared to GA that uses binary representation to present the solution. Figure 2 shows the DE in pseudocode format.

A work that used the DE in fuzzy system was carried out by Dagon and Akgungor [17]. In the study, they used DE to tune the membership function of fuzzy system for controlling traffic light signals. In their work, the fuzzy system contained two levels. The first level will search the minimum and maximum value of fuzzy set, to generate the delay between colors. Meanwhile, the second level was used in reshaping the membership function and optimizing it using the DE. The proposed method was tested with nine traffic scenarios, and the result concluded that the proposed method had significant potential in optimizing the membership function for traffic light signal control. Another work that demonstrated the use of the DE in fuzzy system was done by Pishkenari et. al. in [18]. In their work, they used the ability of the DE in optimizing the design of fuzzy controller in tracking the mobile robot trajectory. They used the DE in tuning the membership function in their method. Their method performed well, and the experimental result showed that the proposed method that used the DE was more acceptable than the other meta-heuristic algorithms like the GA. The implementation of fuzzy system with the DE has also been performed in the biological data mining by [19]. In the study, the fuzzy system was used in predicting lung cancer while DE was utilized in feature selection by eliminating unnecessary attributes in fuzzy rules based on the lung cancer data. The proposed method was compared with the fuzzy classification rule where the rule was taken from the experts. The result showed that the proposed method performed much better and was able to optimize the rules that contained less input of the data.

```
Begin
    Generate randomly an initial population of
    solutions.
    Calculate the fitness of the initial population.
    Repeat
    For each parent, select three solutions at
    random.
        Create one offspring using the DE
        operators.
        Do this a number of times equal to the
        population size.
        For each member of the next generation
            If offspring(x) is more fit than
            parent(x)
              Parent(x) is replaced.
    Until a stop condition is satisfied.
End.
```

Figure 2: The pseudocode of DE

3. SWARM INTELLIGANCE

The works of SI started in the beginning of 1990s where Beni and Wang used the term in their cellular robotic system. Since then, SI has evolved to be used in a group of agents that work independently to achieve a specific goal [20], [21]. To achieve the goal, all agents interact and exchange information with each other through several activities which are inspired from nature including chemical message-carrier (pheromone by ants), dance (waggle dance by bees) and self-organized patterns in the foraging process. The process of SI starts with generation of candidate solution by locating all candidate solutions located randomly in a search space. Then, the best of the candidate solution will be selected based on its fitness in order to improve the quality of another candidate solution. The improvement process is normally inspired by the nature. This process will continue until the termination condition is reached or the maximum number of iterations is attained. There are a lot of SI methods proposed such as Particle Swarm Optimization (PSO) [22], Ant Colony Optimization (ACO) [23], Artificial Bee Colony Optimization (ABC) [24] and Bacterial Foraging Optimization [25]. This work only reviewed two algorithms which are PSO and ACO. This is due to the fact that these two algorithms are the pioneer to the SI method, popular and widely used [26], [27].

3.1. Particle Swarm Optimization

PSO is inspired by the social behavior of bird flocking or fish schooling. PSO was introduced in 1995 by Kennedy and Eberhart [22]. The concept of PSO is inspired by the movement of particle where it is based on the best position located. During the iteration process, when the new improved position is found, the current best position will be replaced. Figure 3 shows the PSO in a pseudocode format.

An example of work that used PSO in the fuzzy system is [28], where the author used the combination on fuzzy with PSO to optimize the marine diesel engine rotation speed control. Firstly, the fuzzy control theory was used to design the controller by representing the theory in the fuzzy sets. Then, PSO was utilized to optimize the fuzzy set by fine-tuning the membership function. Then, simulation was performed to test the proposed method and the finding showed that the proposed method performed well when compared to the fuzzy controller without PSO. Another work that implemented PSO in fuzzy system was done in [29]. In that work, the authors proposed an improved PSO in optimizing the fuzzy parameters by extracting the optimal rule and fine-tune the membership function in the fuzzy system for developing a classifier model. In

the proposed method, a crossover and non-uniform mutation from the GA is implemented into the PSO to update the position of the particle and standardize the velocity. The proposed method was tested in ten benchmark datasets and the results showed that the proposed method performed better compared to other works reported in the literature. Ponce et. al. [30] have implemented PSO with the fuzzy system in brushless direct current motor (BLDCM). In their work, the BLDCM needed to be optimized to operate in high frequency, high temperature, large voltage, and quick changes of current. They used the fuzzy system to model the DC motor operation and the PSO was utilized to tune the model of DC motor in order to achieve the optimum BLDCM system. The simulation was performed to test the proposed method and the result showed that the proposed method was acceptable in optimizing the BLDCM system.

```
Initalize particles population
do
   for each particle p with position x_p do
   calculate fitness value f(x_p)
      if f(x_p) is better than pbest_p then
         pbest_p ← x_p
      endif
   endfor
   Define gbest_p as the best position found so far
   by any of p's neighbors
   for each particle p do
      v_p ← compute_velocity(xp, pbest_p, gbest_p)
      x_p ← update_ position(x_p, v_p)
   endfor
while (Max iteration is not reached or a stop
criterion is not satisfied)
```

Figure 3: The pseudocode of PSO

3.2. Ant Colony Optimization

Ant Colony Optimization (ACO) was introduced by Marco Dorigo in his colleagues in the early 1990s [23]. The search activities mimic the behavior of ant colonies that can jointly figure out the shortest route between their food and their nest without any visual information. Pheromones act as the communication medium and used by ants to interact with each other. Using this concept, Marco Dorigo and his colleagues developed ACO where ants can find the shortest path between nest and food source and applied this on optimization problem in finding the best solution [31]. Figure 4 shows the ACO in a pseudocode format.

A work that used ACO in the fuzzy system has been demonstrated by Juang and Chang in [32]. In their work, they proposed an improved method of the ACO named continuous ACO (RCACO) to design and generate the rules in the fuzzy system. Firstly, the RCACO will decide the number of rules, generate the antecedent parts and consequent part before optimizing the rules. In the proposed method, the antecedent parts and consequent part were represented by the path of ants while the pheromone level decides a new path-selection. The proposed method was compared with other meta-heuristic methods and the result showed that the RCACO performed better. Another example that used ACO in the fuzzy system has been proposed in [33]. The author proposed an improved ACO named adaptive ant colony algorithm (AACA) that hybridizes with fuzzy logic controller in controlling an inverted pendulum. In the proposed method, the fuzzy rules were used to model the inverted pendulum system by the input of fuzzy rules representing the four state variables in the pendulum system. Then, the AACA was utilized in tuning the fuzzy rules by automatically adjusting the strategy in selecting the ant paths of AACA. The simulation was performed and the AACA was very effective in controlling the inverted pendulum system. Kondratenko in his work [34] proposed the modified ACO in optimizing the fuzzy rules of the Mamdani-type fuzzy systems for automatic control system of the reactor temperature system. The proposed method was able to generate the rules based on several conditions which were; not enough initial information, different level of expert knowledge, complex rules such as large number of rules and uncertain expert knowledge. Several computer simulations were performed, and the result showed that the proposed method was very effective in controlling the reactor temperature system.

```
Begin
      Set parameters
      Initialize the pheromone trails scheduled
        activities
        Construction of solutions by ants
        Server of actions (Optional)
        Updating pheromone
End-Scheduled activities
```

Figure 4: The pseudocode of ACO

4. PHYSICS BASED ALGORITHM

The theory of quantum computing system proposed by Richard Feynman in 1982 has opened a new door to the physics-inspired meta-heuristic algorithm. The PBA is a category of algorithm that is motivated from physics laws. There is no standard flow or process of the PBA like in EA and SI that involves recombination, reproduction, and iteration. This is because most of the algorithms in the PBA work based on physics laws such as theorem of gravity, movement of universe, quantum theory, electromagnetism law and river system. There are a lot of PBAs such as Simulated Annealing (SA) [35], Gravitational Search Algorithm (GSA) [36],

Big Bang-Big Crunch Algorithm [37] and Electromagnetism-like Algorithm (EMA) [38]. This study chose SA and GSA. The reasons behind choosing these algorithms are because they are widely used, and these algorithms were the earliest of algorithms to be discovered.

Provide a statement that what is expected, as stated in the "Introduction" chapter can ultimately result in "Results and Discussion" chapter, so there is compatibility. Moreover, it can also be added the prospect of the development of research results and application prospects of further studies into the next (based on result and discussion).

4.1. Simulated Annealing Algorithm

The SA method was found in [35] where they applied it in statistical mechanics to simulate cooling in a heat bath. Kirkpatrick et al. proposed the use of SA in optimization problem [39]. The SA was inspired by the annealing technique, where it involves the process of heating a material and then slowly lowering the temperature to strengthen its volume and decrease its defects. The SA is not a population-based algorithm; thus, it starts with a random initial solution and initial temperature in a search space and becomes the current solution. In each iteration, a new solution is randomly selected in a neighborhood, then the difference between the new solution with the current solution is measured. If the new solution is better, it will replace the current solution. Figure 5 shows the SA in a pseudocode format.

Input: *ProblemSize*, $iterations_{max}$, $temp_{max}$
Output: S_{best}
1: $S_{current} \leftarrow$ CreateInitialSolution(*ProblemSize*)
2: $S_{best} \leftarrow S_{current}$
3: For (i to $iterations_{max}$) do
4: $S_i \leftarrow$ CreateNeighborSolution($S_{current}$)
5: $temp_{curr} \leftarrow$ CalculateTemperature(i , $temp_{max}$)
6: If (Cost(S_i)≤ Cost(S_{best}))
7: $S_{current} \leftarrow S_i$
8: If (Cost(S_i)← Cost(S_{best}))
9: $S_{best} \leftarrow S_i$
10: End
11: ElseIf (Exp $\left\{\frac{(Cost(Scurrent)-Cost(Sbest))}{tempcurr}\right\}$ > *Rand*())
12: $S_{current} \leftarrow S_i$
13: End
14: End
15: Return ()

Figure 5: The pseudocode of SA

4.2. Gravitational Search Algorithm

In 2009, Rashedi et al. [36] introduced GSA based on the law of gravity and the law of mass interactions. The GSA used the concept of every particle in the universe that is attracted to each other by their force, where the force is based on particle mass, and the distance between every particle is contrary to the square of the particle mass. In GSA, the candidate solution is presented by an agent, where all agents are considered as objects (particle) in the universe. The position of the object represents the solution of the problem. The fitness value of the object is measured by its masses. The GSA works when all the objects are attracted to each other by a gravity force and causes all of them to move toward the objects that have heavier masses (the best solution). Figure 6 shows the GSA in a pseudocode format.

In a work proposed by Bardamova et. al. [40], they used GSA to design a fuzzy rule-based classifier. Their proposed method consists of three phases which are feature selection where they used the basis of the binary GSA in feature selection and followed by the construction of fuzzy rule-based classification phase using extreme feature values before applying continuous GSA in optimum parameter identification of fuzzy rule-based parameters. Another work that used GSA in fuzzy modeling was proposed in [41]. They used the fuzzy system method in dealing with a nonlinear system of a control ball and beam system. In their work, they utilized GSA to tune the controller parameters, thus improving the performance of fuzzy system in controlling the ball and beam system. Bala and Malhotra [42] in their work presented a classification system for breast tumor detection using fuzzy system and GSA. In their method, fuzzy system was used in the classification process and GSA was improved and known as comprehensive learning GSA (CLGAS) where the CLGAS worked to find the optimal solution.

1: Search space identification, $t = 0$
2: Randomized initialization $X_i(t)$ for $i = 1,2,\ldots, N$
3: **While** stopping criteria is not satisfied **Do**
4: Evaluate the fitness for each agent
5: Update $G(t)$, $best(t)$, $worst(t)$ and $M_i(t)$ for $i = 1,2, \ldots, N$
6: Calculate of acceleration and velocity
7: Updating agents position to yield $X_i (t + 1)$ for $i = 1,2,\ldots,N$, $t = t + 1$
8: **Endwhile**
9: **Output**: Best solution found.

Figure 6: The pseudocode of GSA

5. SOCIAL BEHAVIOR-BASED ALGORITHM

SBBA can be considered as the latest category of meta-heuristic algorithm. It was inspired based on social human behavior. Similar to PBA, there is no specific flow or process of the SBBA because every algorithm in SBBA has its own process in controlling the search process. There are lots of

SBBA that have been proposed such as Imperialist Competitive Algorithm (ICA) [43], League Championship Algorithm (LCA) [44], Teaching-Learning Based Optimization Algorithm (TLBO) [45], and Socio Evolution and Learning Optimization Algorithm (SELO) [46]. This study focuses on ICA and TLBO because those algorithms can be considered as the earliest and are widely used in the fuzzy system.

5.1. Imperialist Competitive Algorithm

The ICA is an algorithm that is inspired by human social evolution, namely sociopolitical imperialist competition that is proposed by Atashpaz-Gargari and Lucas [43]. The ICA begins with a set of candidate solution generated randomly where the candidate solution is known as a country. Then, the selection process starts based on the cost function of each country to become the imperialists to control the other countries, where the unselected countries are known as colonies. The cost function (fitness) will determine the power of each country. Then, the imperialist will take control of the other colonies to form the initial empires. Two operators of the ICA will be applied in the evolution process, which are assimilation and revolution. The assimilation process is a moving process of the colonies close to the imperialist. Meanwhile, the revolution process is about making random changes of position on colonies. With these operators, a colony can obtain a better position, able to take control of the entire empire and replace the current imperialist with a newfound solution. Figure 7 shows the ICA in a pseudocode format.

A work that implemented ICA in a fuzzy system has been carried out by Nouri et al. [47]. In their work, a hybrid approach has been proposed known as HYEI, in discovering the rule-based system in the fuzzy system. Their approach consists of three stages: the first stage features selection using embedded ICA feature selection, then phase two is generating fuzzy rules for classification and the final stage uses ICA to optimize the rule by reducing the length and number or rules. A study performed by Bagheri and Marj [48] used ICA for optimization of the fuzzy parameter, which is membership function. They utilized the ability of the ICA in tuning membership function of fuzzy system for controlling a robotic arm. The implementation of ICA in their work produced good results where the fuzzy system was able to control the arm better and smoother. Another work that used ICA in optimization of fuzzy system was done in [49]. The authors used ICA in designing a fuzzy controller for enhancing the transient and the behavior of a vehicle system. ICA was implemented to tune the fixed rule in the system based on the membership function of the input and output of the variables. The result of this study indicated that ICA performed better than the expert controller in the optimization process.

1: Generate an initial population randomly and calculate their objective function.
2: Select imperialist states and initialize empires.
3: Move the colonies toward their relevant imperialist.
4: If there is a colony with lower cost than its related imperialist, exchange their position. 5: Calculate the total cost of an empire based on the power of imperialist and its colonies. 6: Pick the weakest colony from the weakest empire and give it to the empire that has the most likelihood to possess it (Imperialistic competition).
7: If there is an empire without colony removes it.
8: If there is one empire left, stop algorithm, if not go to step3.

Figure 7: The pseudocode of ICA

5.2. Teaching-Learning Based Optimization Algorithm

The TLBO was proposed by Rao and Savsani in 2012 [45]. TLBO was inspired from the teaching-learning process based on the effect of influence of a teacher on the output of learner in a class (Rao, 2016). There are two basic modes of learning which are teacher phase and learner phase. Teacher phase means the learning process is through teacher and learner phase means the learning process is through interaction with other learners. In TLBO, a group of learners is considered as a population and different subjects offered to the learners are considered as different design variables of the optimization problem. A learner's result is analogous to the fitness value of the optimization problem. The best solution in the entire population is considered as the teacher. Figure 8 shows the TLBO in a pseudocode format.

A work that used TLBO in a fuzzy system has been presented in [50]. In their work, they presented a fuzzy system-TLBO based in stabilizing a two-link planar horizontal under-actuated manipulator. The fuzzy system has been used to control the system while the TLBO was utilized for searching the optimum of fuzzy parameters in controlling the system. Their method has performed well in stabilizing the system and showed effectiveness and robustness in different conditions. Another work that utilized TLBO in a fuzzy system has been carried out in [51]. The authors presented an improved fuzzy system which was the adaptive fuzzy logic controller (AFLC) with TLBO in controlling the frequency of autonomous AC microgrid. In their method, AFLC was used to control the frequency with input and output of membership function. Then, TLBO improved the performance of AFLC by tuning the membership function because the

autonomous AC microgrid highly depends on the frequency of autonomous. Sahu et al. [52] used fuzzy proportional–integral–derivative (PID) controller in their work to control the automatic generation control (AGC) in a thermal system. They used the ability of TLBO in tuning the parameter in a fuzzy-PID controller. They compared their work with another existing meta-heuristic algorithm such as GA, PSO and SA, and their proposed method performed better.

```
Begin
        Best = pop [1] {optimal}
        Index = 1
        for ( i = 2 to pop_size)
        if Fitness (pop [i] is better than Fitness (pop [i]) )
        Best = pop [i]
        Index = i
        endif
        end for
        Best_Solution = pop [Index]
End
```

Figure 8: The pseudocode of TLBO

6. COMPARATIVE ANALYSIS OF THE META-HEURISTIC ALGORITHMS

In selecting the suitable meta-heuristic algorithms for fuzzy modeling, there are many factors that need to take into account. Usually, many factors will contribute to the performance and ability of the algorithm such as the representation of the fuzzy parameters; the interpretability of the fuzzy model produced by the algorithm; the parameters of algorithm such as number of population, and the specific parameters according to the algorithm itself; the process involves in the algorithm; the processing speed of the algorithm; and the easiness of the implementation of the algorithm. Table 1 lists out a comparative analysis of the meta-heuristic algorithms reported in the literature.

Table 1: Comparative analysis of meta-heuristic methods

Algorithm	Advantages	Disadvantages
Genetic Algorithm	• Efficient and simple programmability. This is because chromosomes can be represented as bit strings. This makes the representation easy and the process to represent the solution is fast [53]. • Can work in parallel/multiple solution which acts as independent agents and each agent can explore the search space simultaneously [54]. • The GA is efficient when the search space is large and complex. In addition, the GA performs well when the knowledge in domain/expert knowledge is scarce and incomplete [55].	• Operating on dynamic data sets is difficult [56], [57]. • Involves many parameters, therefore choosing the value of parameters will affect the performance[53], [54]. • Performs poorly in finding the global optimum, always stuck in local optimum and premature convergence because mutation and crossover are performed randomly [58], [59].
Differential Evolution Algorithm	• The performance of the DE is more robust compared to GA [60]. • DE is fast in finding the solution and requires less computation time if compared to other algorithms in the EA category[58], [61]. • Like GA, DE is easy to be used to represent the solution [62].	• The exploration and exploitation are not well balanced, hence sometimes the DE can be trapped in the local optima and affect the convergence precision and convergence speed [63]. • Noise of the data can affect the performance of DE due to its greedy nature [64]. • The performance depends on the parameter setting, thus it is very difficult to set the parameter value and knowledge is needed to apply the DE for practical applications [65], [66].
Particle Swarm Optimization Algorithm	• The performance of the PSO is fast and easy to be used because it adopts the real number code, and it is decided directly by the solution [27], [67]–[69]. • The PSO is less dependent on a set of initial points[70]–[72]. • The PSO only has a few parameters and it is simple to implement [59], [73]	• Easy to be trapped in local optimum especially with complex problems such as in the case of multi-dimensional and multimodal problems [74], [75]. • The PSO does not perform well in the problem of scattering and non-coordinate system [27], [76], [77]. • The performance of the PSO is very dependent on one of its parameters,

		which is the velocity. It will affect the performance if the value of the velocity is not set properly [78].
Ant Colony Optimization Algorithm	• The ACO has guaranteed convergence which means it is able to find the optimum solution [59], [68]. • The solution in the search space is less redundant because the ACO uses graph in its operation and guarantees the optimal solution can be found [79]. • The ACO is flexible where it can adapt the change to the problem while the ACO is being processed [68].	• The ACO is hard to be applied and its theoretical analysis is difficult to be understood [27], [68]. • The selection value of pheromone parameter in the formula is too empirical and needs knowledge, experience and theoretical argument to maintain the efficiency of convergence [80]. • The processing of the ACO is slow and takes a longer time to find the optimum solution [68].
Simulated Annealing Algorithm	• Calculation process is simple, easy to realize, general, robust, suitable for parallel processing, and can be used to solve complex nonlinear optimization problem [81]. • SA provides a clear and simple approach to find near optimal solutions of di□cult combinatorial optimizations where there are many local minima [82]. • The SA is very stable and balances between local and global search [83].	• The process of the SA is very slow and has a problem in convergence speed [81], [84], [85]. • The search process may terminate in the local minima and the SA is unable to explore the global space because it is based on single solutions [68]. • The SA does not work well on multi-objective optimization [83].
Gravitational Search Algorithm	• GSA is adaptive in its learning rate and memory-less algorithm [86]–[89] • The exploration process of the GSA is very good, thus it performs better in global searches [90], [91]. • The concept of the GSA is easy to understand and this algorithm is simple to use [88].	• Slow convergence and the tendency to become trapped in local minima [92]. • The GSA does not have the mechanism of local search, therefore the ability of local search is weak and convergence speed is not good [93]. • The search process of the GSA is very slow; thus, it will affect the exploitation ability and convergence rate [90], [94].
Imperialist Competitive Algorithm	• The ease of performing neighborhood movement and the ICA does not depend too much on initial point of solutions [95], [96]. • The ICA is simple and easy to use; therefore, it is time saving [97]. • The ICA has good performance of finding optimum solution and has a good convergence speed [98]–[100]. • The ICA has an ability in solving the optimization problems that have many local minima [101].	• ICA is a competition among empires. If the quality improvement approach for empires is not strong enough, competition occurs too often and weak empires get eliminated quickly [96]. • Population diversity quickly degrades and consequently the algorithm is trapped in local optima due to loss of diversity. These negative points may cause premature convergence to a local optimum in the ICA [96]. • The ICA is easy to stuck in the local optima, thus affecting its performance [102].
Teaching-Learning Based Optimization Algorithm	• The TLBO is free-parameter - not requiring any parameter of the algorithm for its operation with the exception of the population size and maximum number of iterations [57], [103], [104] • The TLBO is excellent in global search compared to other meta-heuristic algorithms [103], [105]. • The implementation of the TLBO is easy and simple because it is parameter-free [57], [106].	• The TLBO involves a lot of iterations (in the learning phase and teacher phase), so it is a time-consuming method [107]. • The TLBO consumes lot of computer memory space thus makes it time-consuming [108]. • The TLBO is easy to be trapped in the local optimum due to having no control parameter in measuring the distance of the best student and the teacher, and the

		mean of class [109]. • The TLBO has premature convergence and poor exploiting ability, especially in the problems involving multimodal function [106].

7. CONCLUSION

This paper focused on finding the patterns of fuzzy systems with meta-heuristic algorithms for classification problems. The fuzzy system is seen as a good choice to deal with this problem. Fuzzy system works by implementing the fuzzy logic and approximate reasoning. In developing the fuzzy system, fuzzy parameters are needed to identify in order to obtain the desired behavior of the system. This process is known as fuzzy modeling. Fuzzy works well in simple problems. However, when applied on complex problems, the construction of fuzzy becomes complicated. This might be due to the identification of parameters. Due to that, a method is needed for identifying the fuzzy parameters with an automatic process. This paper presented an overview of algorithms to automate fuzzy modeling by identifying the fuzzy parameters. In the presented paper, eight algorithms that are widely used in fuzzy modeling for classification methods have been presented and discussed in detail, where the discussion covered the pseudo-code, strengths and weaknesses. The algorithms chosen are GA, DE, PSO, ACO, SA, GSA, ICA and TLBO. Based on the discussion of these algorithms, it is hard to determine which method is the best one because each has its own advantages and disadvantages. The selection of method depends on the problem and features selected because there is no single method that works best on every problem

ACKNOWLEDGEMENTS

Special appreciations to Universiti Malaysia Pahang for the sponsorship of this study by approve the Ministry of Higher Education (MOHE) for Fundamental Research Grant Scheme (FRGS) with Vot No. RDU190113.

REFERENCES

1. R. A. Rahim, N. Othman, M. N. S. Zainudin, N. A. Ali, and M. M. Ismail, **Iris Recognition using Histogram Analysis via LPQ and RI-LPQ Method**, *International Journal of Electrical & Computer Sciences*, vol. 12, no. 4, pp. 66–70, 2012.
2. R. Feraud and F. Clerot, **A methodology to explain neural network classification**, vol. 6080, no. April, 2002.
3. L. Auria and R. A. Moro, **Support vector machines (SVM) as a technique for solvency analysis**, *DIW Berlin Discussion Paper*, pp. 1–16, 2008.
4. M. G. Tsipouras, T. P. Exarchos, and D. I. Foriadis, **A methodology for automated fuzzy model generation**, pp. 1–2, 2008.
5. M. A. Bin Ismail, **A fuzzy cooperative genetic algorithm for fuzzy modeling mohd arfian bin ismail universiti teknologi malaysia**, 2011.
6. J. H. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press, 1975.
7. L. J. Fogel, **Autonomous Automata**, *Industrial Research*, vol. 4, no. 2, pp. 14–19, 1962.
8. I. Rechenberg, **Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution**. Stuttgart, Germany: Frommann-Holzboog Verlag, 1973.
9. R. Storn and K. Price, **Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces**, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
10. J. R. Koza, D. Andre, F. H. Bennett, and M. A. Keane, **Genetic Programming III: Darwinian Invention {&} Problem Solving. San Francisco**, CA, USA: Morgan Kaufmann Publishers Inc., 1999.
11. C. K. Chong, M. S. Mohamad, S. Deris, M. S. Shamsir, L. E. Chai, and Y. W. Choon, **Parameter Estimation by Using an Improved Bee Memory Differential Evolution Algorithm (IBMDE) to Simulate Biochemical Pathways**, *Current Bioinformatics*, vol. 9, no. 1, pp. 65–75, 2014.
12. L. L. Jiang, D. L. Maskell, and J. C. Patra, **Parameter estimation of solar cells and modules using an improved adaptive differential evolution algorithm**, *Applied Energy*, vol. 112, pp. 185–193, Dec. 2013.
13. L. G. M. de Souza, H. Haida, D. Thévenin, A. Seidel-Morgenstern, and G. Janiga, **Model selection and parameter estimation for chemical reactions using global model structure**, *Computers {&} Chemical Engineering*, vol. 58, pp. 269–277, 2013.
14. M. A. Ismail, H. Asmuni, and M. R. Othman, **The fuzzy cooperative genetic algorithm (FCoGA): The optimisation of a fuzzy model through incorporation of a cooperative coevolutionary method**, *Journal of Computing*, vol. 3, no. 11, pp. 81–90, 2011.
15. A. Plerou, E. Vlamou, and V. Papadopoulos,

Fuzzy Genetic Algorithms : Fuzzy Logic Controllers and Genetics Algorithms, *Global Journal For Research Analysis*, vol. 5, no. 11, pp. 497–500, 2016.
16. C. Chen, M. Li, J. Sui, K. Wei, and Q. Pei, **A genetic algorithm-optimized fuzzy logic controller to avoid rear-end collisions**, *Journal of Advanced Transportation*, vol. 50, no. 8, pp. 1735–1753, 2016.
17. E. Doğan and A. P. Akgüngör, **Optimizing a fuzzy logic traffic signal controller via the differential evolution algorithm under different traffic scenarios**, *SIMULATION*, vol. 92, no. 11, pp. 1013–1023, 2016.
18. H. Nejat Pishkenari, S. H. Mahboobi, and A. Alasty, **Optimum synthesis of fuzzy logic controller for trajectory tracking by differential evolution**, *Scientia Iranica*, vol. 18, no. 2 B, pp. 261–267, Apr. 2011.
19. S. Poongothai, C. Dharuman, P. Venkatesan, and A. Professor, **Application of Fuzzy Differential Evolutionary Algorithms in Biological Data Mining**, *International Journal of Scientific & Engineering Research*, vol. 10, no. 1, pp. 1379–1382, 2019.
20. M. Dorigo, M. Birattari, and T. Stutzle, **Ant colony optimization**, *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.
21. M. Patil, J. Irani, and V. Jagtap, **Survey Of Different Swarm Intelligence Algorithms**, *International Journal of Advance Engineering and Research Development*, vol. 1, no. 12, pp. 2348–4470, 2014. https://doi.org/10.21090/IJAERD.011217
22. J. Kennedy and R. Eberhart, **Particle Swarm Optimization**, in *Proceedings of IEEE International Conference on Neural Networks*, 1995, vol. 4, pp. 1942–1948.
23. M. Dorigo, V. Maniezzo, and A. Colorni, **Ant system: Optimization by a colony of cooperating agents**, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
24. D. Karaboga and B. Basturk, **A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm**, *Journal of Global Optimization*, vol. 39, no. 3, pp. 459–471, Apr. 2007.
25. K. M. Passino, **Biomimicry of bacterial foraging for distributed optimization and control**, *IEEE Control Systems Magazine*, vol. 22, no. 3, pp. 52–67, 2002.
26. L. Lhotská, M. Macaš, and M. Burša, **PSO and ACO in optimization problems**, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 4224 LNCS, pp. 1390–1398.
27. V. Selvi and R.Umarani, **Comparative Analysis of Ant Colony and Particle Swarm Optimization Techniques**, *International Journal of Computer Applications*, vol. 5, no. 4, pp. 1–6, 2010.
28. T. A. Tran, **The optimization of marine diesel engine rotational speed control process by fuzzy logic control based on Particle Swarm Optimization algorithm**, *Future Internet*, vol. 10, no. 10, 2018.
29. P. Ganeshkumar, C. Rani, and S. N. Deepa, **Formation of fuzzy if-then rules and membership function using enhanced particle swarm optimization**, *International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems*, vol. 21, no. 1, pp. 103–126, 2013.
30. P. Ponce, L. Arturo Soriano, A. Molina, and M. Garcia, **Optimization of Fuzzy Logic Controllers by Particle Swarm Optimization to Increase the Lifetime in Power Electronic Stages**, in *Electric Machines for Smart Grids Applications - Design, Simulation and Control*, A. El-Shahat, Ed. 2018.
31. R. Jangra and R. Kait, **Analysis and comparison among Ant System; Ant Colony System and Max-Min Ant System with different parameters setting**, in *3rd IEEE International Conference on* , 2017.
32. C. F. Juang and P. H. Chang, **Designing fuzzy-rule-based systems using continuous ant-colony optimization**, *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 1, pp. 138–149, 2010.
33. B. Zhao, **Optimal design for fuzzy controller by ant colony algorithm**, *Key Engineering Materials*, vol. 439–440, pp. 1190–1196, 2010.
34. Y. P. Kondratenko and A. V. Kozlov, **Generation of rule bases of fuzzy systems based on modified ant colony algorithms**, *Journal of Automation and Information Sciences*, vol. 51, no. 3, pp. 4–25, 2019.
35. N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, **Equation of state calculations by fast computing machines**, *Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1090, 1953.
36. E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, **GSA: A Gravitational Search Algorithm**, *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.
37. [37]H. M. Genç, I. Eksin, and O. K. Erol, **Big bang - Big crunch optimization algorithm hybridized with local directional moves and application to target motion analysis problem**, in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, 2010, pp. 881–887.
38. Ş. I. Birbil and S. C. Fang, **An**

electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, vol. 25, no. 3, pp. 263–282, Mar. 2003.

39. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, **Optimization by simulated annealing**, *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
40. M. Bardamova, A. Konev, I. Hodashinsky, and A. Shelupanov, **A fuzzy classifier with feature selection based on the gravitational search algorithm**, *Symmetry*, vol. 10, no. 11, 2018. https://doi.org/10.3390/sym10110609
41. M. J. Mahmoodabadi and N. Danesh, **Gravitational search algorithm-based fuzzy control for a nonlinear ball and beam system**, *Journal of Control and Decision*, vol. 5, no. 3, pp. 229–240, 2018.
42. I. Bala and A. Malhotra, **Fuzzy classification with comprehensive learning gravitational search algorithm in breast tumor detection**, *International Journal of Recent Technology and Engineering*, vol. 8, no. 2, pp. 2688–2694, 2019.
43. E. Atashpaz-Gargari and C. Lucas, **Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition**, in *2007 IEEE Congress on Evolutionary Computation, CEC 2007*, 2007, pp. 4661–4667.
44. A. H. Kashan, **League Championship Algorithm: A new algorithm for numerical function optimization**, in *SoCPaR 2009 - Soft Computing and Pattern Recognition*, 2009, pp. 43–48.
45. R. V. Rao, V. J. Savsani, and D. P. Vakharia, **Teaching-Learning-Based Optimization: An optimization method for continuous non-linear large scale problems**, *Information Sciences*, vol. 183, no. 1, pp. 1–15, Jan. 2012.
46. M. Kumar, A. J. Kulkarni, and S. C. Satapathy, **Socio evolution & learning optimization algorithm: A socio-inspired optimization methodology**, *Future Generation Computer Systems*, vol. 81, pp. 252–272, Apr. 2018.
47. D. Jalal Nouri, M. Saniee Abadeh, and F. Ghareh Mohammadi, **HYEI: A new hybrid evolutionary imperialist competitive algorithm for fuzzy knowledge discovery**, *Advances in Fuzzy Systems*, vol. 2014, 2014.
48. M. Bagheri and H. F. Marj, **A Fuzzy Logic Controller tuned with Imperialist competitive algorithm for 2 DOF robot trajectory control with help Scaling factor**, *Research Journal of Fisheries and Hydrobiology*, vol. 10, no. 9, pp. 53–57, 2015.
49. A. M. Jasour, E. Atashpaz, and C. Lucas, **Vehicle fuzzy controller design using imperialist competitive algorithm**, in *Second First Iranian Joint Congress on Fuzzy and Intelligent Systems, Tehran, Iran*, 2008, no. February, pp. 1–6.
50. M. J. Mahmoodabadi and A. Yazdizadeh Baghini, **Design of an optimal fuzzy controller of an under-actuated manipulator based on teaching-learning-based optimization**, *Acta Mechanica et Automatica*, vol. 13, no. 3, pp. 166–172, 2019.
51. A. Anilkumar and N. V. Srikanth, **Teaching-learning optimization based adaptive fuzzy logic controller for frequency control in an autonomous microgrid**, *International Journal of Renewable Energy Research*, vol. 7, no. 4, pp. 1842–1849, 2017.
52. B. K. Sahu, S. Pati, P. K. Mohanty, and S. Panda, **Teaching-learning based optimization algorithm based fuzzy-PID controller for automatic generation control of multi-area power system**, *Applied Soft Computing Journal*, vol. 27, pp. 240–249, Aug. 2015.
53. A. Meyer-Baese and V. Schmid, **Chapter 5 - Genetic Algorithms**, in *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, Second Edi., A. Meyer-Baese and V. Schmid, Eds. Oxford: Academic Press, 2014, pp. 135–149.
54. X. Yang, **Nature-Inspired Metaheuristic Algorithms**, in *Nature-Inspired Metaheuristic Algorithms Second Edition*, Luniver Press, 2010, p. 115.
55. A. Venkateswara, G.A.V.Ramachandra, and M. V. Basaveswara, **Coping and limitations of genetic algorithms**, *Oriental Journal of Computer Science & Technolog*, vol. 1, no. 2, pp. 137–141, 2008.
56. B. S and S. S. S, **A Survey of Bio inspired Optimization Algorithms**, *International Journal of Soft Computing and Engineering*, vol. 2, no. 2, pp. 137–151, 2012.
57. B. Mohanty and S. Tripathy, **A teaching learning based optimization technique for optimal location and size of DG in distribution network**, *Journal of Electrical Systems and Information Technology*, vol. 3, no. 1, pp. 33–44, May 2016.
58. N. Karaboga and B. Cetinkaya, **Performance Comparison of Genetic and Differential Evolution Algorithms for Digital FIR Filter Design**, in *Advances in Information Systems*, 2005, pp. 482–488.
59. M. N. Ab Wahab, S. Nefti-Meziani, and A. Atyabi, **A comprehensive review of swarm optimization algorithms.**, *PloS one*, vol. 10, no. 5, p. e0122827, Jan. 2015.
60. S. T. Suganthi, D. Devaraj, and S. H. Thilagar, **An improved differential evolution algorithm for congestion management considering voltage stability**,

Indian Journal of Science and Technology, vol. 8, no. 24, 2015.
61. A. D. Lilla, M. A. Khan, and P. Barendse, **Comparison of Differential Evolution and Genetic Algorithm in the design of permanent magnet Generators**, in *Proceedings of the IEEE International Conference on Industrial Technology*, 2013, pp. 266–271.
62. U. K. Rout, R. K. Sahu, and S. Panda, **Design and analysis of differential evolution algorithm based automatic generation control for interconnected power system**, *Ain Shams Engineering Journal*, vol. 4, no. 3, pp. 409–421, Sep. 2013.
63. W. Xiang, X. Meng, M. An, Y. Li, and M. Gao, **An Enhanced Differential Evolution Algorithm Based on Multiple Mutation Strategies**, *Computational Intelligence and Neuroscience*, vol. 2015, p. 285730, 2015. https://doi.org/10.1155/2015/285730
64. S. Binitha and S. S. Sathya, **A Survey of Bio inspired Optimization Algorithms**, no. 2, pp. 137–151, 2012.
65. Z. Yang, X. Yao, and J. He, **Making a Difference to Differential Evolution**, in *Advances in Metaheuristics for Hard Optimization*, P. Siarry and Z. Michalewicz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 397–414.
66. X. Wang and S. Zhao, **Differential Evolution Algorithm with Self-Adaptive Population Resizing Mechanism**, *Mathematical Problems in Engineering*, vol. 2013, p. 419372, 2013.
67. W. Zhuo and X. Yu, **A Particle Swarm Optimization Algorithm Based on Dynamic Adaptive and Chaotic Search**, in *IOP Conference Series: Materials Science and Engineering*, 2019, vol. 612, no. 5.
68. Z. Abdmouleh, A. Gastli, L. Ben-Brahim, M. Haouari, and N. A. Al-Emadi, **Review of optimization techniques applied for the integration of distributed generation from renewable energy sources**, *Renewable Energy*, vol. 113, pp. 266–280, Dec. 2017.
69. S. S. Aote, M. M. Raghuwanshi, and L. Malik, **A Brief Review on Particle Swarm Optimization: Limitations & Future Directions**, *International Journal of Computer Science Engineering*, vol. 2, no. 5, pp. 196–200, 2013.
70. D. Choubey, S. Paul, K. Bala, M. Kumar, and D. U. Singh, **Implementation of a Hybrid Classification Method for Diabetes**, in *Intelligent Innovations in Multimedia Data Engineering and Management*, 2018, pp. 201–240.
71. K. Y. Lee and J. Park, **Application of Particle Swarm Optimization to Economic Dispatch Problem: Advantages and Disadvantages**, in *2006 IEEE PES Power Systems Conference and Exposition*, 2006, pp. 188–192.
72. S. Das, A. Abraham, and A. Konar, **Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives**, in *Studies in Computational Intelligence Volume*, vol. 116, 2008, pp. 1–38.
73. B. S. G. de Almeida and V. C. Leite, **Particle Swarm Optimization: A Powerful Technique for Solving Engineering Problems**, in *Swarm Intelligence*, J. Del Ser, E. Villar, and E. Osaba, Eds. Rijeka: IntechOpen, 2019.
74. H. Wang and Z. Zhou, **A Heuristic Elastic Particle Swarm Optimization Algorithm for Robot Path Planning**, *Information*, vol. 10, no. 99, 2019.
75. M. Li, W. Du, and F. Nian, **An Adaptive Particle Swarm Optimization Algorithm Based on Directed Weighted Complex Network**, *Mathematical Problems in Engineering*, vol. 2014, p. 434972, 2014.
76. D. P. Rini, S. M. Shamsuddin, and Siti Sophiyati Yuhaniz, **Particle Swarm Optimization: Technique, System and Challenges**, *International Journal of Computer Applications*, vol. 14, no. 1, pp. 19–27, 2011.
77. Q. Bai, **Analysis of Particle Swarm Optimization Algorithm**, *Computer and Information Science*, vol. 3, no. 1, pp. 180–184, Jan. 2010.
78. P. Szynkiewicz, **A Comparative Study of PSO and CMA-ES Algorithms on Black-box Optimization Benchmarks**, *Journal of Telecommunications and Information Technology*, vol. 8, pp. 1–13, 2018.
79. J. Ning, C. Zhang, P. Sun, and Y. Feng, **Comparative Study of Ant Colony Algorithms for Multi-Objective Optimization**, *Information*, vol. 10, no. 1, p. 11, Dec. 2018.
80. Wang, Guanyu, **A Comparative Study of Cuckoo Algorithm and Ant Colony Algorithm in Optimal Path Problems**, *MATEC Web Conf.*, vol. 232, p. 3003, 2018.
81. L. Lin and C. Fei, **The Simulated Annealing Algorithm Implemented by the MATLAB**, *International Journal of Computer Science Issues*, vol. 9, no. 6, pp. 357–360, 2012.
82. R. Diekmann, R. Lüling, and J. Simon, **Problem Independent Distributed Simulated Annealing and its Applications**, in *Applied Simulated Annealing*, R. V. V Vidal, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 17–44.
83. K. Amine, **Multiobjective Simulated Annealing: Principles and Algorithm Variants**, *Advances in Operations Research*, vol. 2019, p. 8134674, 2019.

84. P. Lv, L. Yuan, and J. Zhang, **Cloud theory-based simulated annealing algorithm and application**, *Engineering Applications of Artificial Intelligence*, vol. 22, no. 4–5, pp. 742–749, Jun. 2009.
85. L. Ingber, **Simulated annealing: Practice versus theory**, *Mathematical and Computer Modelling*, vol. 18, no. 11, pp. 29–57, Dec. 1993.
86. S. Duman, Y. Sonmez, U. Guvenc, and N. Yorukeren, A**pplication of gravitational search algorithm for optimal reactive power dispatch problem**, in *INISTA 2011 - 2011 International Symposium on INnovations in Intelligent SysTems and Applications*, 2011, pp. 519–523.
87. S. Nagpal, S. Arora, S. Dey, and S. Shreya, **Feature Selection using Gravitational Search Algorithm for Biomedical Data**, in *Procedia Computer Science*, 2017, vol. 115, pp. 258–265.
88. S. J. Safi, S. S. Tezcan, I. Eke, and Z. Farhad, **Gravitational Search Algorithm (GSA) based PID Controller Design for Two Area Multi-Source Power System Load Frequency Control (LFC)**, *Gazi University Journal of Science*, vol. 31, no. 1, pp. 139–153, Mar. 2018.
89. R. K. Sahu, S. Panda, and S. Padhan, **Optimal gravitational search algorithm for automatic generation control of interconnected power systems**, *Ain Shams Engineering Journal*, vol. 5, no. 3, pp. 721–733, Sep. 2014.
90. H. Hu, X. Cui, and Y. Bai, **Two Kinds of Classifications Based on Improved Gravitational Search Algorithm and Particle Swarm Optimization Algorithm**, *Advances in Mathematical Physics*, vol. 2017, p. 2131862, 2017.
91. J. Liu, Y. Xing, and Y. Li, **A gravitational search algorithm with adaptive mixed mutation for function optimization**, *International Journal of Performability Engineering*, vol. 14, no. 4, pp. 681–690, Apr. 2018.
92. D. Shen, T. Jiang, W. Chen, Q. Shi, and S. Gao, **Improved chaotic gravitational search algorithms for global optimization**, in *2015 IEEE Congress on Evolutionary Computation, CEC 2015 - Proceedings*, 2015, pp. 1220–1226.
93. T. Chengyi, **Gravitational Search Algorithm Based on Simulated Annealing**, *Journal of Convergence Information Technology*, vol. 9, no. 2, pp. 231–237, 2014.
94. Y. Kumar and G. Sahoo, **A Review on Gravitational Search Algorithm and its Applications to Data Clustering & Classification**, *Intelligent Systems and Applications*, vol. 6, pp. 79–93, 2014.
95. Z. Ardalan, S. Karimi, O. Poursabzi, and B. Naderi, **A novel imperialist competitive algorithm for generalized traveling salesman problems**, *Applied Soft Computing Journal*, vol. 26, pp. 546–555, Jan. 2015.
96. Y. Abdi, M. Lak, and Y. Seyfari, **GICA: Imperialist competitive algorithm with globalization mechanism for optimization problems**, *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 25, no. 1, pp. 209–221, 2017. https://doi.org/10.3906/elk-1507-226
97. E. Rezaei, A. Karami, and M. Shahhosseni, **The Use of Imperialist Competitive Algorithm for the Optimization of Heat Transfer in an Air Cooler Equipped with Butterfly Inserts**, *Australian Journal of Basic and Applied Sciences*, vol. 6, no. 3, pp. 293–301, 2012.
98. H. Kia, S. H. Ghodsypour, and H. Davoudpour, **A hybrid imperialist competitive algorithm for solving economic lot and delivery scheduling problem in a four-stage supply chain**, *Advances in Mechanical Engineering*, vol. 9, no. 2, p. 168781401668689, Feb. 2017.
99. E. Atashpaz Gargari, F. Hashemzadeh, R. Rajabioun, and C. Lucas, **Colonial competitive algorithm: A novel approach for PID controller design in MIMO distillation column process**, *International Journal of Intelligent Computing and Cybernetics*, vol. 1, no. 3, pp. 337–355, Aug. 2008.
100. T. You, Y. Hu, P. Li, and Y. Tang, **An improved imperialist competitive algorithm for global optimization**, *Turkish Journal of Electrical Engineering & Computer Sciences*, vol. 27, pp. 3567–3581, 2019.
101. H. Emami, S. Dami, and H. Shirazi, **K-Harmonic Means Data Clustering with Imperialist Competitive Algorithm**, *UPB Scientific Bulletin, Series C: Electrical Engineering*, vol. 77, no. 1, pp. 91–104, 2015.
102. S. Wang, Aorigele, W. Kong, W. Zeng, and X. Hong, **Hybrid Binary Imperialist Competition Algorithm and Tabu Search Approach for Feature Selection Using Gene Expression Data**, *BioMed Research International*, vol. 2016, p. 9721713, 2016.
103. B. Amiri, **Application of Teaching-Learning-Based Optimization Algorithm on Cluster Analysis**, *Journal of Basic and Applied Scientific Research*, vol. 2, no. 11, pp. 11795–11802, 2012.
104. R. Venkata Rao and G. G. Waghmare, **A comparative study of a teaching-learning-based optimization algorithm on multi-objective unconstrained and constrained functions**, *Journal of King Saud University - Computer and Information Sciences*, vol. 26,

no. 3, pp. 332–346, Sep. 2014.

105. X. Kong, Z. Li, D. Zou, and G. Pan, **Novel Teaching-learning-based Optimization Algorithm for Design of Digital Filters**, *{IOP} Conference Series: Materials Science and Engineering*, vol. 611, p. 12031, Oct. 2019.
106. Z. Wu and R. Xue, **A Cyclical Non-Linear Inertia-Weighted Teaching–Learning-Based Optimization Algorithm**, *Algorithms*, vol. 12, no. 5, p. 94, May 2019.
107. L. S. de Oliveira, T. Yoneyama, and L. R. Rodrigues, **Parameter Tuning of The Teaching-Learning Based Optimization Algorithm Applied to Troubleshooting**, in *Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, 2016, no. September, pp. 1988–1999.
108. A. García-Monzó, H. Migallón, A. Jimeno-Morenilla, J.-L. Sánchez-Romero, H. Rico, and R. Rao, **Efficient Subpopulation Based Parallel TLBO Optimization Algorithms**, *Electronics*, vol. 8, no. 1, p. 19, Dec. 2018. https://doi.org/10.3390/electronics8010019
109. S. Talataharia, V. Goodarzimehra, and N. Taghizadieh, **Hybrid Teaching-Learning-Based Optimization and Harmony Search for Optimum Design of Space Trusses**, *Journal of Optimization in Industrial Engineering*, vol. 13, no. 1, pp. 177–194, 2020.