Optimizing Complex Adaptive Systems

Nasser Shebka †

Northern Border Univesity, Arar, KSA

Abstract
Many complex adaptive systems proposed models that attempt to utilize more than two problem solving tools or techniques such as fuzzy logic, machine learning, and genetic algorithms usually involve combining at least two techniques in one module, examples of such combinations are found in techniques such as machine learning using genetic algorithms, fuzzy machine learning, or fuzzy genetic algorithms. A tradeoff must be done between the combined technique's expected problem solving capability and between harvesting each individual technique's capability. We argue that, while integrating these methods may not significantly guarantee an increase of the ability of such systems in problem solving, but may also increase their complexity in a manner that represents a challenge for any optimization attempt. The narrow problem scope that these systems target also presents an objective that we attempted to address here. In this paper we proposed a novel algorithmic approach to optimize complex adaptive systems by emphasizing on their modularity property through segregating the used techniques into phases. We attempted to demonstrate the validity of our method by proposing a model consisting of four parts as follow: a fuzzy logic controller, a cluster-based adaptive genetic algorithm, an unsupervised machine learning algorithm, and the final component is a supervisory optimization algorithm that combines tuning modifiers of the parameters responsible for determining the overall results of the model's other three components. The model's resulting extreme complexity is due to its objective to cover a broad range of problem spaces and not pre-defined situations. We concluded that the modularity and adaptation of our presented model offers a promising and challenging unexploited territory of complex adaptive systems and their optimization attempts that require further exploration.

Key words:
Complex adaptive systems, fuzzy logic controller, cluster-based adaptive genetic algorithms, fuzzy C-means clustering

1. Introduction and Motivation

From Complex adaptive systems (CASs) share four major features: parallelism, conditional action, modularity, and adaptation and evolution as identified by J. Holland. The latter two translate to recombination and competition which are mechanisms considered significantly in our proposed model. CASs behavior becomes difficult to predict since they tend to use internal models to anticipate the future while the expected outcomes determine the current actions. Another contributing factor is the lack of relevant theories lead to failure to identify CASs "lever points" [1]. Attempts to model human reasoning processes in adaptive problem solving settings usually limits the scope of any problem, and even then categorizes the model as a CAS with certain features in order to qualify as an evolving system . Our proposed model is an attempt in a direction that has always intrigued scientists, that is to produce a general problem solving model by proposing a system that can address a wider range of problems, as it frequently faces infinite problems that hinder the possibility of producing such systems. Most relevant endeavors in academia and in industry are usually tailored to particular problems, fields, situations, or cases of semi-defined environments in best case scenarios. Any attempt to broaden of the problem's scope will exponentially increase the complexity of the required representational tool. However, the motivation of the proposed system is driven by the generalization problem which can be further explained by the principle of incompatibility as proposed by Zadeh. The principle implies that as the complexity of a system increases; its predictability diminishes until a threshold is reached beyond which precision and significance become almost mutually exclusive. Furthermore, optimizing a generalized solution is very much characterized by: complexity, uncertainty, demand of rapid information acquirement, and the immediate processing of the acquired information, thus justifying the selection and construction of the proposed system's components the way they are presented in this paper. Another major challenge is the ability to exactly represent any system using a mathematical model regardless of being able of describing it in simple linguistic rules. The main reasons for choosing the algorithms used in our model are; the flexibility they offer in knowledge representation, the wide range of problems they can address, and most importantly, their compatibility with each other and in coherence with objective on which the proposed model is constructed. In the following sections and before presenting the proposed model's structure and functionality, each of the used algorithms are introduced and discussed along with some detailed aspects explain the reasons for selecting them. Our

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.59

proposed model is composed of three parts with a supervising algorithm. The initial phase is a FLC, the second is a cluster-based adaptive GA (CAGA), and the third is an UL phase that uses a fuzzy c-means FCM clustering algorithm. The SOA consists of multiple modifiers. This paper is organized as follows:

i. The first section is introduction and motivation. It includes an extensive analysis of each of the used techniques and methods and reasons of selecting very specific techniques within the broader method, this section namely includes: fuzzy logic controllers and rule based systems, adaptive genetic algorithms, machine learning and fuzzy C-means clustering.
ii. The second section is a literature review of related research.
iii. In this third section we introduced the proposed model starting with the overall structure of the system. Each of the model's four components is dedicated a segment that explains the used technique in details, and also describes the operation method and relation and integration to other components.
iv. The fourth section of the paper discusses optimization of the proposed model describing specific and detailed optimization parameters to be considered in the functionality of the system. This section also emphasis on issues and challenges facing the model's performance.
v. The final section includes the conclusion and future work.

1.1 Fuzzy logic controllers and rule based systems

Fuzzy logic controllers (FLCs) are Knowledge based systems that attempt to mimic human expert knowledge in dealing with and controlling nonlinear systems through a mathematical forms using fuzzy sets and fuzzy logic that was proposed by Zadeh in 1968. Fuzzy systems are an optimization tool that has the ability to model complex systems through its capability of searching through wide range of variables driven by some of their unique ability to use linguistic variables instead of numerical variables. Simplicity while being capable of representing relations between variable through fuzzy conditional statements is another advantage. Additionally, there's the ability to characterize complex relations through fuzzy algorithms. Fuzzy systems deal with imprecision and uncertainty usually for control or classification modeling of problems as a result to the non-linearity output. The widely used fuzzy models are known as fuzzy rule based systems. They are a set of logical fuzzy rules, or any classical rule based systems that deal with fuzzy (IF-THEN) rules. Fuzzy rule based systems are used for solving search or optimization problems and extend to modeling, control, data mining, and classification problems[2]. There are several types of fuzzy rule based systems but the two main types are: the Mamadani-type, which produces a class or an output action, and the Tagaki-Sugeno-Kang, which produces a polynomial function [3]. The introduction of fuzzy computation to conventional rule based systems improved these systems capability of addressing optimization problems under the strain of fuzzy or uncertain data formulated by the (IF-THEN) rules composed of fuzzy logic statements [4].

1.2 Adaptive genetic algorithms

Genetic algorithms (GAs) were introduced by Holland in 1973. They are an optimization technique that utilizes the natural selection mechanism. They provide an efficient tool for exhaustive search in complex spaces and use a fitness function repeatedly on a set of strings called population consisting of randomly generated finite strings called individuals until the desired output is reached [5]. The fitness function is an evaluating benchmark of the proximity of a given solution to the optimum or desired solution. All possible solutions are treated as competing individuals. The typical process of a GA can be briefly explained in the following [6]:

A pair of parent chromosomes is selected from the current population according to the fitness function. The selected pair undergoes a crossover process forming two offspring according to a crossover rate probability. The offspring are mutated according to a mutation rate probability, and then placed in a new population. Finally, the initial population is replaced with the new one.

GAs require a genetic representation of the solution domain and a fitness function to evaluate that domain according to which the initial selection is performed [5]. However, the best solutions are measured in comparison to others from the generated pool. This in turn yields an issue of local optimization and autonomy. This can be considered as an advantage or disadvantage depending on how the GA is integrated to the overall system and the contextual use of the GA in the specified model or nature of the problem[7]. The concept of adaptive GAs was proposed by Goldberg in 1989 [5] and numerous scholars began exploring adaptive GAs by changing the clan, the selection process, or the crossover and mutation probabilities in order to improve the state of premature convergence and to avoid losing qualitative chromosomes. In a cluster–based adaptive genetic algorithm (CAGA) a population's state of optimization is measured depending on clustering analysis [8], but more importantly , both parameters of crossover and mutation probabilities that greatly determine the degree of rule accuracy are flexible and make it easier to use different optimization methods with GAs, such as fuzzy systems.

1.3 Machine learning and fuzzy C-means clustering

Machine learning (ML) mainly uses various complex statistical models to achieve some prediction objective that is usually determined by the nature of the problem or involved data. It's an excellent optimization tool given an existing data set derived from experience and depends mainly on modeling and optimization as the used techniques. The training data determines the nature of the final result or optimal solution. Unsupervised machine learning (UL) increases the autonomy of any model as it has the ability to classify data with no benchmark labels. This property justifies our choice to utilize it in the proposed model. Fuzzy C-means (FCM) clustering is a type of UL techniques. It is a widely used clustering method in which any element (data point, rule) can belong to more than one cluster with a membership degree. It is very similar to fuzzy K-means clustering algorithm but with the addition of a membership values *wij* and a fuzzifier *m* parameters [9]. The full inverse distance weighting is done as every point is evaluated with each cluster. There are two functional stages required from the clustering algorithms:

- Predict which rules should be clustered together.
- Learn from the previous prediction (provided its validity by the environment) and update its rule base accordingly.

2. Related work

Complex problem solving systems using hybrid approaches that consist of two or more of computational Intelligence techniques such as FLCs, GAs, and UL in a rule-based driven systems attracted considerable attention both in literature and in industry, but when considering a general problem space optimization approach with a segregated UL algorithm, the research tends to become a bit scarce. However, several researches proposed a combination of three of the aforementioned tools with the alteration of implementing machine learning; in the GA portion of the proposed model , based on the Michigan Approach Classifier System , based on the Pittsburgh approach [10], and based on the Iterative rule learning approach [11]. There is considerable research regarding adaptive systems specially in relation to the tools used in the proposed model such as; adaptive GAs by tuning the crossover and mutation probabilities [8], adaptive FLCs [12], or even designing adaptive FLCs using GAs . Additionally , Using adaptive GAs to solve clustering problems were introduced into literature through various forms, in particular, those that use K-means algorithms facing challenges like difficulties of capturing global optimal solution and issues of avoiding local minima of K-means [13; 14]. Numerous models of complex Adaptive GA models were introduced along with alterations and dependencies of GA's parameters, such as; real-coded GA (RGA), binary-coded GA (BGA) , Boeringer's time-varying mutation [15; 16], and adaptive clustering-base GA (ACGA) as an example of limited problem scope applicability [17].

3. The Proposed model

3.1 System's structure

The proposed model as shown in Fig. 1 consists of four parts, three phases and a supervising algorithm. The initial phase is a FLC, the second is a cluster-based adaptive GA (CAGA), and the third is an UL phase that uses a FCM clustering algorithm. The SOA consists of modifiers of the following parameters: membership function, scaling function, fitness function, elitist function, mutation probability, and crossover probability. All modifiers are interconnected to an evaluation and validation controller that is linked with the external environment in which the problem situation occurs. Segregating the GA from UL phase is to benefit from UL capabilities of providing strong domain independent search method for learning tasks. Although UL is not a learning algorithm in itself, but the use of an UL method that relies significantly on exploring patterns using C-means algorithm is proceeded by using a similar algorithm to adjust the crossover and mutation probabilities *px* and *pm* in the CAGA portion of the model [8].

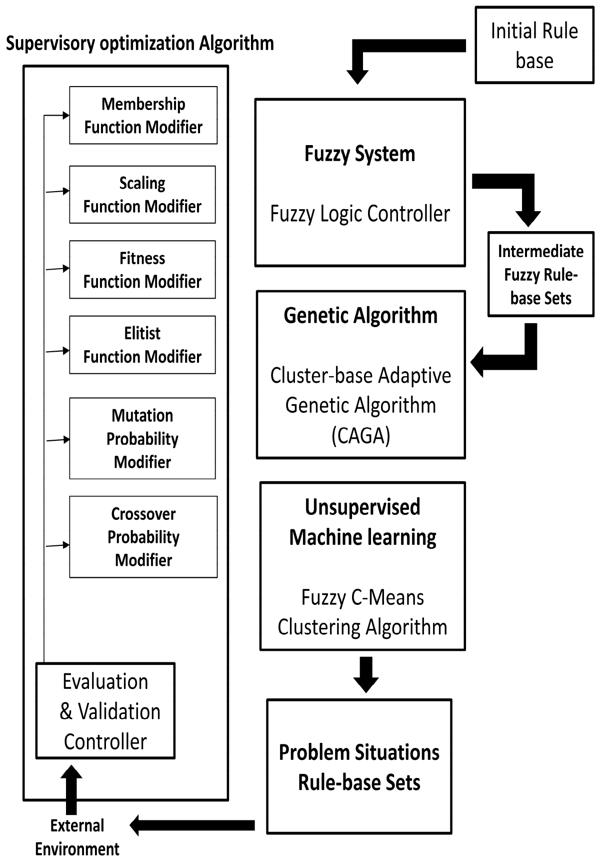

Fig. 1 System's structure.

3.2 The Fuzzy logic controller

For the FLC we used a Mamdani-type fuzzy rule base system. The reason for using this inference method is that it produces fuzzy sets as the output of membership function. This method also considers the combinations of multiple conditions that reflect real life situations, which require non-linear modeling [1]. As shown in Fig. 2, the FLC can be viewed as a knowledge-based system consisting of the following components: a fuzzy data base, a fuzzy rule base as initial repositories, fuzzification, inference, and defuzzification stages. Nevertheless, our primary focus is on optimizing the FRBSs though it is important to acknowledge the effect of aggregate fuzzy databases in formulating and optimizing the rules of the FRBSs. The reason for such acknowledgement is that the database usually consist of the scaling functions definitions of variables and fuzzy sets linguistic labels associated membership function. Building intelligent initial rule bases with flexible answers is achieved by relating fuzzy logic programming with fuzzy control as explained in [18] with the exception of not confining our system to fuzzy Herbrand interpretations [19]. The initial rule base consists of two parts: the antecedents, and the consequences. While each rule construct of the resulting FRBSs consists of three parts: the antecedents, the consequences, and the connective. This enables us to further consider the potential of all possible combinations even if some combinations may seem implausible or unlikely. This can be further addressed through the use of one of the multi-objective fuzzy systems categories described in [20]. The FLC uses a triangular membership function for simplicity purposes. Another main reason is that triangular membership functions allow rapid system response[21].

.

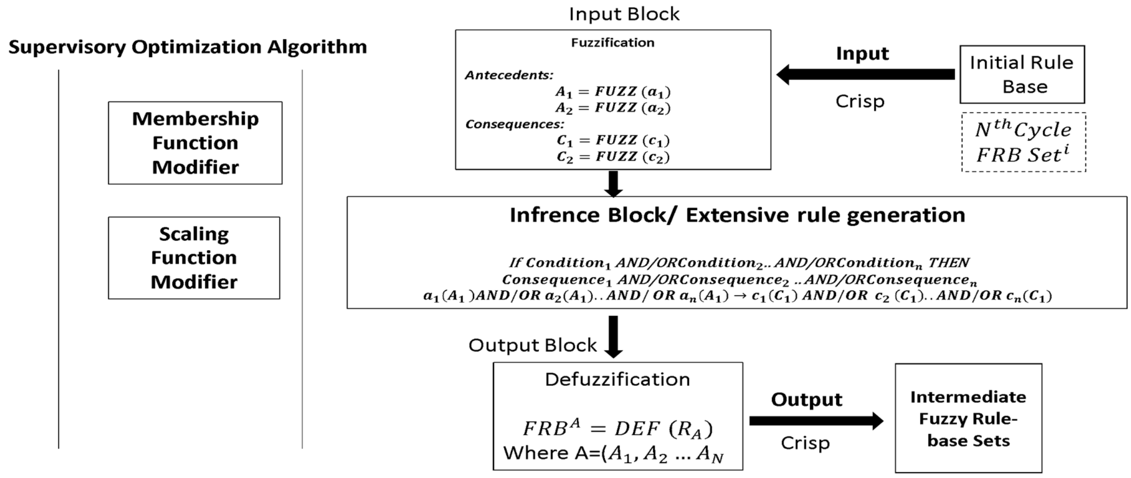

Fig. 2 The fuzzy logic controller

Each FRBS is oriented around a particular situation that corresponds to the conditions of each of the antecedents in a permutation like relation with all the possible and viable consequences. This can be expressed in fuzzy sets terms by placing each fuzzy rule conjunction in a Cartesian product space, hence, placing every rule in a different universe of discourse. Although we considered the simple form of the fuzzy (IF-THEN) rules, but we can argue that expanding the “fuzziness” of these rules to take the following form:

$$\mathbf{If\ (A_{1\,is}\ a)\ AND/OR\ (A_2\ is\ b)\ THEN\ (C_1\ is\ c)\ AND/OR\ (C_2\ is\ d)} \quad (1)$$

Where A = (A_1 , A_2, ... , A_N) and C=(C_1 , C_2 , ... , C_N)

But as we can observe, this it is not quite exhaustive. Henceforth, we then considered the following form instead:

$$\mathbf{If\ (A_{1\,is}\ a_1)\ AND/OR\ (A_1\ is\ a_2)..AND/OR\ (A_1\ is\ a_n)\ THEN\ (C_1\ is\ c_1)\ AND/OR\ (C_1\ is\ c_2)..AND/OR\ (C_1 is\ c_n)}$$

$$\vdots$$

$$\mathbf{If\ (A_N\ is\ a_1)\ AND/OR\ (A_N\ is\ a_2)..AND/OR\ (A_N\ is\ a_n)\ THEN\ (C_N\ is\ c_1)\ AND/OR\ (C_N\ is\ c_2)..AND/OR\ (C_N is\ c_n)}$$

Translating into

$$\mathbf{a_1(A_1)\ AND/OR\ a_2(A_1)..AND/\ OR\ a_n(A_1) \rightarrow c_1(C_1)\ AND/OR\ c_2\ (C_1)..AND/OR\ c_n(C_1)} \quad (2)$$

3.3 The Cluster-based adaptive genetic algorithm

The resulting intermediate FRBSs are the input of the succeeding CAGA phase which combines both advantages of adaptation and clustering methods. The CAGA is mainly a selection and clustering method of the elite candidates from the resulting population of the FLC. Separating the CAGA from ML process ensures the quality of the resulting elite candidates and allows for further refining through the following UL phase. This method is similar to an Adaptive GA (AGA) method proposed in [22] in which the best chromosomes are obtained and then undergo the K-means algorithm as higher quality clustering results while in our model it is done repetitively. Operation wise, the CAGA’s main objective is to produce the local optima for the resulting FRBSs, and when combined with the swapping process of adaptive mutation resulting in super set as the initial pre-defined FRBS for the SOA tuning process modifier. The clustering of each of the populations (FRBSs) using the K-means algorithm can only be used to partition clusters that have already been partially optimized to some degree [23], and this is previously done in the FLC phase. After cross over and mutation, the resulting offspring are ranked and only the fittest is chosen as part of the fuzzy solution set through an elitist function, while the elitist selection is used to avoid the shortcomings of the mutation rate value that can either lead to genetic drift, premature convergence of the CAGA, or the loss of good solutions. The algorithm's steps as shown in Fig. 3 are as follows:

1. Initialization: each FRBS corresponding to a problem situation and therefore containing all similar rules pertaining to that particular situation is already combined as resulting output of the previous FLC algorithm and consist of fuzzy rules such as:

$$\mathbf{If}\ (A_{1\,\text{is}}\, a_1)\ \mathbf{AND/OR}\ (A_1\ \mathbf{is}\ b_2)..$$
$$\mathbf{If}\ (A_{1\,\text{is}}\, b_1)\ \mathbf{AND/OR}\ (A_1\ \mathbf{is}\ c_5)..$$
$$\mathbf{If}\ (A_{1\,\text{is}}\, y_{11})\ \mathbf{AND/OR}\ (A_1\ \mathbf{is}\ z_n)..\quad (3)$$

2. $p_{m,}$ p_x are also initialized , while the maximum number of generations G_{max} is initialized only in the first cycle of the system but determined by the SOA afterwards.
3. Chromosomes selection: fuzzy rules with highest fitness value are selected for swapping with other clusters, since the main concern here is not to eliminate good and viable rules.
4. Reproduction: each of the swapped rules will be reproduced by the crossover and mutation operations with elite members of the new FRB Set.
5. The Elitist function: for each generation; the elite member will be recorded and compared with that of the previous generation and each of the generations to ensure that only best of all generations are selected.
6. Steps 2, 3, and 4 are repeated for the determined G_{max}.
7. Adaptive control of crossover and mutation probabilities p_m and p_x is done using a fuzzy system in the OSA.

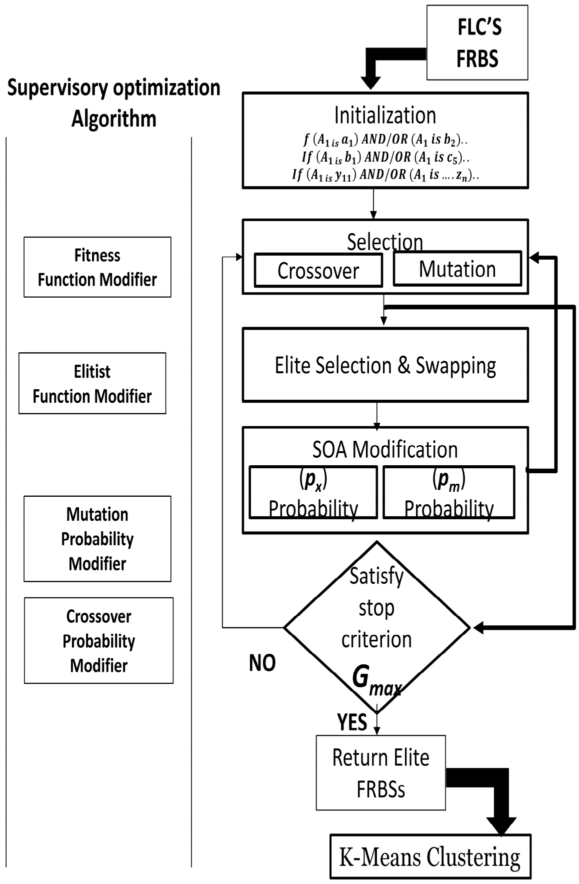

Fig. 2 The Cluster-based adaptive genetic algorithm process

Although the use of an elite function implies its convergence to a single rule, however, a mechanism to allow for the identification of multiple optimal rules should be considered to maintain the diversity of the FRBSs since the adaptive elitist technique can be used with typical genetic operators. For the swapping process, elite rules from a FRBS replace the worst rules of another FRBS only if their fitness is higher. Clustering is then performed on the resulting scattered fuzzy rules from the previous phase to produce clusters that correspond to situations using a fuzzy K-means algorithm, and the process is as follows:

a) Randomly, the number of clusters k is set (predefined)

b) K random points are selected from the data as centroids (Initial clusters centers are chosen ($FRBS^1, FRBS^2, \ldots, FRBS^k$) randomly).
c) All the closest points to the cluster centroids are labeled as such.
d) The centroids of newly formed clusters are recomputed as the mean of all the points of the cluster.
e) Steps 3 and 4 are repeated until there is no change in centroids values.

As for the termination condition of the CAGA, it consists of a combination of the highest ranking rules insinuated by using the elitist method, after which successive iterations only produce inferior rules and a fixed number of generations for efficiency considering the number of FRBSs. The number of generation is fixed only per cycle and is determined and altered by the SOA according to a relational function of the number of FRBSs and each population size.

3.4 The Fuzzy C-means algorithm

The third phase of the model entails an UL process characterized by a FCM clustering algorithm which provides a coherent tool considering the final objective of the model, which is to re-cluster the scattered FRBSs resulting from the previous phase in a manner that will ensure that the new clusters reflect the complexity and exhaustiveness of real life potential situations that can occur, with minimum rule redundancies if not at all. Representation of the relative assigning of rules according to their degree of necessity to each of these situations/clusters, with the sum of coefficients for the points degree of being the k^{th} cluster u^k (x) is achieved by the following formula:

$$\forall x \sum_{k=1}^{Num.of clustrs.} u_k = 1 \quad (4)$$

Where a number of clusters is randomly chosen, coefficients are randomly assigned to each rule as a degree of membership to each cluster, and finally, a repeated process of computing the centroid of each cluster and the coefficient of each rule is computed as a degree of its belonging to that cluster/situation. The centroid is computed using the formula:

$$C_k = \frac{\sum_r w_k(r)^m r}{\sum_r w_k(r)^m} \quad (5)$$

Where m is the fuzzifier that determines the clusters fuzziness level such that $m \in R$ and $m \geq 1$, meaning that a greater m will result in a smaller membership values wij which are the addition of the FCM objective function it aims to minimize over the K-means algorithm, that is:

$$\arg min_C \sum_{i=1}^{n} \sum_{j-1}^{c} w_{ij}^{m} \left\|R_i - C_j\right\|^2 \quad (6)$$

Where:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{\|R_i - C_j\|}{\|R_i - C_k\|} \right)^{\frac{2}{m-1}}}$$

and m is set to 2 if any domain knowledge or feedback is available from the external environment which is rarely the case since it defies the purpose of adding an SOA that handles that. However, the FCM has the same limitation of the fuzzy k-means algorithm in that the minimum is a local minimum, so the initial assigning on weights will determine the results, but with a crucial qualitative difference that while it is slower than the fuzzy k-means method, but every point is weighted against every cluster determined by measurement of inverse distance to the cluster's center. For simplicity, the value of K is determined by the value and number of antecedents, such that: *(A_1 is a_1) (A_1 is a_2)... (A_1 is a_n)*, as well as their conjoining operators (AND, OR) instead of the usual elbow, hierarchical, or dendogram methods [24], although it can similarly be determined using a combination of mixed antecedents and consequences but within a controlled rule formats.

3.5 The Supervisory optimization algorithm (SOA)

The SOA as shown in Fig. 1 is designed to adjust parameters and functions of two of the three phases of the model; the FLC and the CAGA. The SOA can be viewed as an optimization driven tuning algorithm that uses feedback from the environment given that the case is that each of the addressed parameters can achieve the optimal output depending on various factors relating to problem's nature, situation, and/or particular circumstances (if any) during solution attempt/s.

4. System optimization

4.1 Tuning the scaling function and membership function

The non-linear scaling function consists of six contraction parameters that are tuned by the SOA to optimize the matching between the variable range and the universe of discourse for every FRBS that has been normalized by the scaling function within the FLC. These parameters correspond to the constructs of the fuzzy rules:

$$3\ \boldsymbol{Anticident\ parameters}\text{: } \left(\boldsymbol{A_N} - \boldsymbol{a_n} - \boldsymbol{AND/OR(O_A)}\right)$$

$$3\ \boldsymbol{Consquence\ parameters}\text{: } \left(\boldsymbol{C_N} - \boldsymbol{c_n} - \boldsymbol{AND/OR(O_c)}\right)$$

The tuning of the triangular membership function concentrates on determining the length of the FRBSs members in relation to a specific situation of the problem, and although tuning of the membership function in less complicated systems is time consuming [25], but the system's autonomy is a primary consideration. Therefore, in conjunction with the role of the scaling function tuning modifier, the membership function consists of two parameters: length of the antecedent and length of the consequence. The FRBS rules structure is determined previously by the scaling function and tuned by its modifier. Naturally, the initial construction of FRBSs is exhaustive and flawed but rectified in the following cycles by the membership function modifier within the SOA using the feedback from the FCM algorithm and the external environment.

4.2 Tuning the genetic algorithm parameters

As for the tuning of the fitness function of the CAGA which is used to enhance the quality of FRB clusters, a different value is presented to each of FRBSs. This is done taking into consideration adjusting mutation process parameters which greatly influences obtaining superior results through an adaptive mutation operator. The tuning of the crossover and mutation probabilities is fuzzy controlled using environment feedback from the SOA's corresponding modifiers. This allows for the solutions to surpass the local optima as well as enhance the GA's rate of convergence [8]. However, p_m and p_x are evolution state dependent, and therefore should be adapted accordingly. The assigned fitness function to all rules within each of FRB cluster sets is the only specific knowledge used for a given problem, although a problem's domain calculated parameters set function can be used as a fitness function, otherwise, the search for an optimization problem becomes an approximation given that the elitist method measures only the rules within each cluster but the swapping process increases the uncertainty of a local optimization of the cluster or subpopulation. For these reasons, the following modifiers are used with their purpose:

- A fitness function modifier to tune it depending on the feedback from the evaluation and validation controller and external environment.
- An elitist function modifier that determines the number of generations of the CAGA per each cycle function formula.
- Two modifiers to adjust the value of mutation and crossover parameters: probability(number of FRs being mutated) and magnitude (the degree of mutation i.e. number of bits) considering that mutation is the GA's principle operation affecting the degree of the populations diversity [26].

The tuning of the fitness function depends to a great extent on the feedback from evaluation and validation controller and the external environment since it is problem dependent. The fitness function for each FRBS is tuned in coherence with the elitist function, which is tuned to increase or decrease the number of fuzzy rules corresponding to multiple optima within each cluster in addition to finding new unmapped peaks. Adaptation of the probabilities of crossover and mutation is mainly to ensure the quality of generated FRBSs by preventing the elimination of strong fitted genes or rules. The CAGA process sequence can be further reconsidered, particularly the tuning of p_x and p_m by the corresponding modifier as shown in Fig. 3.

4.3 The Evaluation and validation controller

This unit represents the connecting point with the external environment and is responsible for evaluating each rule and validate that it is viable in reality given the extensive rule generation of the FLC.

Although both the CAGA and FCM algorithm phases involve testing rules against some criteria and dropping them, but the number of rules still remains relatively significant yet justified due to real life situations complexity. Optimization and generalization require that the FCM is the only component to interact with the external environment and gives its feedback through the SOA component. This method focuses on generating all potential viable rules from the FLC and enhancing their quality through a simple CAGA process. Tuning of the FLC's membership function indirectly modifies the probability of selection and mutation of each of the CAGA FRBSs, thus, maximizing the FLC output in terms of cluster sizes, while taking into consideration the viability of the generated fuzzy rules and moderately minimizing the resulting CAGA FRBSs qualitative population sizes.

5. Conclusions and Recommendations

In this paper we introduced an approach that emphasizes on the modularity property of complex adaptive systems. The approach segregates problem solving techniques and tools into separate phases rather than integrate them. The objective of our segregation method is to demonstrate that combining or integrating techniques and tools does not necessarily benefit the performance of such systems. On the contrary, it may needlessly increase the system's complexity in a manner that can be challenging when attempting to optimize these systems. The proposed algorithmic model is composed of four parts as follow: a fuzzy logic controller to generate exhaustive fuzzy rules, a cluster-based adaptive genetic algorithm to filter and enhance the quality of the exhaustively generated fuzzy rules and cluster them into fuzzy rule base sets based on an adaptive elitist function, and an unsupervised machine learning algorithm consisted of a fuzzy c-means clustering algorithm to re-cluster the fuzzy rule base sets according to their relation degree to a real life situation characterized by each cluster, and the final component is a supervisory optimization algorithm that combines tuning modifiers of the parameters responsible for determining the overall results of the model's other three components. The expected results by the model are promising, although practically constructing such a model can be very challenging. The first of our findings is that: although there are various tools that can be used to represent, solve, and optimize a wide range of problems through complex adaptive systems, but there are no guidelines on how to proceed when attempting to more than two of these tools or how to combine them as the overall system's output becomes more complex and less predictable. This in turn reflects the extensive shortage level of research regarding complex adaptive systems. The second finding of this paper is that: by further expansion of the proposed model through recombination and refinement, the possibility to achieve higher levels of accuracy and autonomy becomes more likely, but only after all potential formations of the proposed knowledge representation tools have been exploited. Finally, our endeavor in demonstrating the potentials of modularity in complex adaptive systems recommends that further examination of the potentials of all possible combinations composed of different compatible tools, algorithms, and techniques used in constructing complex adaptive systems.

Acknowledgments

The author gratefully acknowledge the approval and the support of this research study by grant no. COM-2017-1-8-F-7470 from the Deanship of Scientific Research at Northern Border University, Arar, K.S.A.

Compliance with ethical standards

Conflict of interest

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

[1] Holland, J. H. 2006. Studying Complex Adaptive Systems *Journal of Systems Science and Complexity*,*19*,1-8. https://doi.org/https://doi.org/10.1007/s11424-006-0001-z
[2] Ishibuchi, H., Nakashima, T. Nii, M. 2004. *Classification and modeling with linguistic information granules: advanced approaches to linguistic data mining.* Springer. https://doi.org/https://doi.org/10.1007/b138232
[3] Takagi, T.Sugeno, M. Feb. 1985. https://doi.org/10.1109/TSMC.1985.6313399. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.*, *SMC-15*(1), 116–132. https://doi.org/https://doi.org/10.1109/TSMC.1985.6313399

[4] Herrera, F. 2008. Genetic fuzzy systems: taxonomy, current research trends and prospects. *Evolutionary Intelligence*, *1*(1), 27–46. https://doi.org/https://doi.org/10.1007/s12065-007-0001-5

[5] Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley. https://doi.org/http://www2.fiit.stuba.sk/~kvasnicka/Free%20books/Goldberg_Genetic_Algorithms_in_Search.pdf

[6] Martinez-Soto, R., Castillo, O., Aguilar, L. T. Melin, P. 2010. Fuzzy logic controllers optimization using genetic algorithms and particle swarm optimization. Advances in Sof Computing: 9th Mexican International Conference on Artifcial Intelligence Proceedings MICAI 2010, , Pachuca, Mexico November 8-13. https://doi.org/10.1007/978-3-642-16773-7_41.

[7] Kivijärvi, J., Fränti, P. Nevalainen, O. 2003. Self-Adaptive Genetic Algorithm for Clustering. *Journal of Heuristics*, *9*, 113-129. https://doi.org/https://doi.org/10.1023/A:1022521428870

[8] Zhang, J., Chung, H. Lo, W. L. 2007. Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms. *IEEE Transactions on Evolutionary Computation*, *11*(3), 326-335. https://doi.org/https://doi.org/10.1109/TEVC.2006.880727.S2CID2625150

[9] Ghosh, S.Kumar, S. 2013. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *International Journal of Advanced Computer Science and Applications*, *4*. https://doi.org/https://doi.org/10.14569/IJACSA.2013.040406

[10] Smith, S. F. 1980. A Learning System Based On Genetic Adaptive Algorithms. In *PhD thesis*. Pennsylvania, USA: Department of Computer Science, University of Pittsburgh.

[11] Venturini, G. 1993. SIA: A supervised inductive algorithm with genetic search for learning attributes based concepts. *Machine Learning ECML-93 Lecture Notes in Computer Science*, 280–296. https://doi.org/https://link.springer.com/chapter/10.1007/3-540-56602-3_142

[12] Herrera, F., Lozano, M. Verdegay, J. L. 1995. Tuning fuzzy logic controllers by genetic algorithms. *International Journal of Approximate Reasoning*, *12*(3-4), 299-315. https://doi.org/https://doi.org/10.1016/0888-613X(94)00033-Y

[13] Hruschka, E. R., Campello, R. J. Freitas, A. A. 2009. A Survey of Evolutionary Algorithms for Clustering. *IEEE Trans. Syst. Man Cybern.*, *39*, 133-155. https://doi.org/https://doi.org/10.1109/TSMCC.2008.2007252

[14] Maulik, U.Bandyopadhyay, S. 2000. Genetic Algorithm-Based Clustering Technique. *Pattern Recognition*, *33*, 1455-1465. https://doi.org/https://doi.org/10.1016/S0031-3203(99)00137-5

[15] Haupt, R. L. 2000. *Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors* IEEE Antennas and Propagation Society International Symposium. https://doi.org/10.1109/APS.2000.875398,

[16] Boeringer, D. W.Werner, D. H. 2002. Adaptive mutation parameter toggling genetic algorithm for phase-only array synthesis. *Electronics Letters*, *38*(25), 1618-1619. https://doi.org/https://doi.org/10.1049/el:20021149

[17] He, T., Li, D. Yoon, S. W. 2018. An adaptive clustering-based genetic algorithm for the dual-gantry pick-and-place machine optimization. *Adv. Eng. Informatics*, *37*, 66-78. https://doi.org/https://doi.org/10.1016/j.aei.2018.04.007

[18] Gerla, G. 2001. Fuzzy control as a fuzzy deduction system. . *Fuzzy Sets and Systems*(121), 409-425. https://doi.org/https://doi.org/10.1016/S0165-0114(00)00124-X

[19] Gerla, G. 2005. Fuzzy Logic Programming and Fuzzy Control. . *Studia Logica* (79), 231-254. https://doi.org/https://doi.org/10.1007/s11225-005-2977-0

[20] Fazzolari, M., Alcalá, R., Nojima, Y., Ishibuchi, H. Herrera, F. 2013. A Review of the Application of Multiobjective Evolutionary Fuzzy Systems: Current Status and Further Directions. *IEEE Transactions on Fuzzy Systems*, *21*(1), 45–65. https://doi.org/https://doi.org/10.1109/TFUZZ.2012.2201338

[21] Zhao, J.Bose, B. K. 2002. *Evaluation of membership functions for fuzzy logic controlled induction motor drive* IEEE 2002 28th annual Conference of the Industrial Electronics Society,, Seville, Spain. https://doi.org/10.1109/IECON.2002.1187512

[22] Zhou, X., Miao, F. Ma, H. 2018. Genetic Algorithm with an Improved Initial Population Technique for Automatic Clustering of Low-Dimensional Data. *Information*, *9*(101). https://doi.org/https://doi.org/10.3390/info9040101

[23] Duda, R., Hart, P. Stork, D. 2001. *Pattern Classification* (Vol. 2nd ed.). Wiley. https://doi.org/https://www.researchgate.net/publication/228058014_Pattern_Classification

[24] Malik , A.Tuckfield, B. 2019. *Applied Unsupervised Learning with R : Uncover hidden relationships and patterns with k-means clustering, hierarchical clustering, and PCA.* Packt Publishing Limited.

[25] Fang, G., Kwok, N. Ha, Q. M. 2008. *Automatic fuzzy membership function tuning using the rarticle swarm optimization* IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application, Wuhan, China. https://doi.org/10.1109/PACIIA.2008.105

[26] Sun, T., Liu, C., Hsieh, S., Lin, C. Lee, K. 2006. *Cluster-based Adaptive Mutation Mechanism To Improve the Performance of Genetic Algorithm* Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06), Jian, China. https://doi.org/10.1109/ISDA.2006.123.

Nasser Shebka received his Masters and PhD degrees in Artificial Intelligence from Al Neelian University, Sudan, where he started his academic career a teaching assistant. He is currently maintain a career in both fields of academia as an associate professor, and professionally as a systems analyst. He also holds an MBA degree from University of Khartoum in addition to several professional certifications in IT, such as CCNA, Oracle OCP, and ITIL v.3. His research interests include fuzzy logic, machine learning, genetic algorithms, and evolutionary knowledge representation tools.