

Research Article

Fine-Tuning the Fuzziness of Strong Fuzzy Partitions through PSO

Ciro Castiello^{*}, Corrado Mencar

Department of Informatics, University of Bari "Aldo Moro", Via Orabona, 4, Bari, 70125, Italy

ARTICLE INFO

Article History

Received 28 Apr 2020

Accepted 01 Sep 2020

Keywords

Strong fuzzy partition
Trapezoidal fuzzy sets
Fuzzy rule-based classifier
Defuzzification
Particle swarm optimization

ABSTRACT

We study the influence of fuzziness of trapezoidal fuzzy sets in the strong fuzzy partitions (SFPs) that constitute the database of a fuzzy rule-based classifier. To this end, we develop a particular representation of the trapezoidal fuzzy sets that is based on the concept of cuts, which are the cross-points of fuzzy sets in a SFP and fix the position of the fuzzy sets in the Universe of Discourse. In this way, it is possible to isolate the parameters that characterize the fuzziness of the fuzzy sets, which are subject to fine-tuning through particle swarm optimization (PSO). In this paper, we propose a formulation of the parameter space that enables the exploration of all possible levels of fuzziness in a SFP. The experimental results show that the impact of fuzziness is strongly dependent on the defuzzification procedure used in fuzzy rule-based classifiers. Fuzziness has little influence in the case of winner-takes-all defuzzification, while it is more influential in weighted sum defuzzification, which however may pose some interpretation problems.

© 2020 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

1. INTRODUCTION

The design of fuzzy inference systems promotes interpretability as a key factor to express the embedded knowledge in a plain readable and understandable way. As a matter of fact, interpretability is the most important quality that justifies the adoption of fuzzy inference systems in real-world applications [1,2,3,4]. When such systems have to be acquired through data-driven approaches, two main design issues arise: (i) the resulting fuzzy inference system should adequately fit data; (ii) the knowledge base should be interpretable to end-users. This led to the development of design methodologies that take into account both accuracy and interpretability [5,6,7,8,9,10,11]; in parallel, the very concept of interpretability, its definition and assessment are matter of current research [12,13,14,15].

In this paper we focus on a specific type of fuzzy inference system, namely the fuzzy rule-based classifier, which adopts a knowledge base consisting of a set of fuzzy classification rules in the form

Rule r : IF x_1 is $A_1^{(r)}$ AND \dots AND x_m is $A_m^{(r)}$ THEN $c^{(r)}$

where a complex antecedent consists of a conjunction of soft constraints binding each variable x_i to a linguistic term $A_i^{(r)}$, and a simple consequent represents a class label $c^{(r)}$. (Further details on fuzzy rule-based classifiers can be found in literature [16,17,18].) The linguistic terms, bound to the same variable in all rules, form a linguistic variable, which includes information to map each linguistic term to a fuzzy set through an operation of interpretation. In

other words, the (explicit) semantics of a linguistic term is defined by a fuzzy set, whose membership function is usually determined through some data-driven process. On the other hand, a linguistic term is usually drawn from natural language (e.g. "low", "old", etc.), therefore it carries an implicit semantics that, in a given context, is assigned by a user when reading the term. The ultimate objective of interpretability-driven design of linguistic variables is to define fuzzy sets so that explicit and implicit semantics of the corresponding linguistic terms highly overlap [19].

The interpretation of linguistic terms determines a collection of fuzzy sets: a standard approach to impose interpretability of linguistic variables is to consider such an ensemble of fuzzy sets as a granulation (or fuzzy partition) of the variable domain. In particular, a convenient way to define such fuzzy sets is through *strong fuzzy partitions* (SFP), i.e., collections of complementary fuzzy sets where the sum of the membership degrees over all fuzzy sets is always equal to the unity, whatever the variable value in the domain. SFPs are convenient tools because they help in satisfying a number of basic interpretability constraints and enable efficient inference since complementarity avoids that many rules are simultaneously active for a given input [20]. Usually, the fuzzy sets of a SFPs are defined through triangular or trapezoidal membership functions, although less common alternatives exist [21,22,23]. In this work we focus on trapezoidal fuzzy sets because they can be efficiently computed and offer greater flexibility than triangular partitions, the latter being considered as a special case.

Typical data-driven design techniques use optimization methods to set the parameters of the fuzzy sets so as to adapt to data. (Eventually, such optimization is constrained to preserve the

^{*}Corresponding author. Email: ciro.castiello@uniba.it

well-formedness of the involved SFPs.) In most cases, fuzzy sets are optimized in terms of both their position and their fuzziness. The first is related to the sub-regions of the domain where the fuzzy sets mostly influence the inference; the latter indicates how much the fuzzy sets are different from crisp sets. Fuzziness and position can be differently measured [24,25]. For the sake of the present study, position can be defined in terms of 0.5 cut; fuzziness can be defined in terms of the slopes of the oblique sides of the trapezoidal fuzzy sets. In this way, it is possible to well separate position and fuzziness, and analyze them separately.

In this work we address the problem of assessing the impact of fuzziness in the performance of a fuzzy rule-based classifier, provided that its position is kept fixed. We consider this issue to be a very important one because many data-driven design methods mainly focus on position of fuzzy sets, while fuzziness is relegated to a more marginal role. As an example, in some works SFPs are defined by looking at prototypes (often obtained after some optimization process), then fuzzy sets of triangular shape are defined so as to form a SFP [26,27,28,29]: in such a case, the information coming from prototypes actually settles the fuzzy sets position, while the determination of fuzziness is only functional to preserve the well-formedness of the partition.

In a previous work of ours, we showed that trapezoidal fuzzy sets offer more degrees of freedom than triangular fuzzy sets, but they still preserve the well-formedness of SFPs [30]. The adoption of triangular fuzzy sets is therefore self-limiting as it introduces a bias which may hinder the reach of acceptable accuracy levels. On the other hand, trapezoidal fuzzy sets require the setting of more hyper-parameters, therefore some existing design techniques (which compute the prototypes of fuzzy sets) cannot be directly applied. To overcome this problem, data-driven optimization can be split into two steps: optimization of position and fine-tuning of fuzziness. We already proposed two methods for fine-tuning the fuzziness of trapezoidal fuzzy sets [31]. Those methods, however, cannot guarantee that the space of all possible SFPs is thoroughly spanned in search of an optimal solution. In this paper we propose an extension of those methods and we prove that this extension is capable to explore the entire search space. In this way, we are able to empirically assess the effects of fuzziness on the overall accuracy of a fuzzy rule-based classifier (while position is kept unchanged), with different inference settings.

After a brief outline of the related work reported in Section 2, in Section 3 we formally define the concepts of SFPs and trapezoidal fuzzy sets, together with the constraints that such fuzzy sets must fulfill to preserve the well-formedness of a SFP. In Section 4 we introduce PSO and we sketch the way to represent trapezoidal SFPs as particles. In Section 5 we formally prove that the proposed representation allows PSO to span the entire space of SFPs with fixed positions. Section 6 summarizes the experimental results on synthetic and benchmark datasets. Finally, Section 7 concludes the work by highlighting the main results and setting the direction of future research.

2. RELATED WORK

The design of SFPs in fuzzy modeling is long-stated. The simplest design approach is uniform granulation, where the domain of a

feature (assumed to be a closed interval in the real line) is partitioned by detecting a number of equally-spaced points in the domain, which eventually serve as prototypes of triangular fuzzy sets. The number of such prototypes is usually user-defined although some computational methods have been proposed in literature for its determination [32].

Noticeably, the position and fuzziness of fuzzy sets determined by uniform granulation do not depend on data: the reasons to use this method are related to the appreciable interpretability of the resulting partitions and the relative difficulty to determine data-driven SFPs with comparable interpretability level. Nevertheless, uniform granulation is too rigid as it does not adapt to data. For such a reason, several methods have been proposed to enable a data-driven design of fuzzy partitions, especially in the realm of evolutionary computation [33].

Starting from some pioneering works on data-driven structure identification of fuzzy rule bases [34], a huge literature developed around the problem of data-driven design of fuzzy partitions [5,35,36,37]. Two general approaches can be identified when attempting to classify data-driven design methods for fuzzy partitions [38]: First Interpretability Then Accuracy (FITA), where fuzzy partitions are first defined (often through uniform granulation) then fine-tuned to data in order to maximize accuracy; First Accuracy Then Interpretability (FATI), where fuzzy partitions are firstly generated from data (often through data clustering) then adjusted to meet a number of interpretability requirements. In most cases, fuzzy sets are adjusted both in their position *and* fuzziness in order to optimize some objective function.

In some cases, fuzziness is explicitly addressed separately from position. As an example, Alcalá *et al.* [39] use a particular representation of linguistic terms involving two different parameters expressing the variation of position and fuzziness with respect to a reference SFP. Specifically, fuzziness is quantified as the length of the support of triangular fuzzy sets. These variations are used as parameters to be optimized through a genetic algorithm (GA). Results show that by adjusting both position and fuzziness, the accuracy of the resulting fuzzy rule-based systems can be improved. That is paid in terms of a decreased interpretability since the resulting partitions are no longer strong. Noticeably, the reported experiments show that the introduction of fuzziness optimization leads to a slight improvement with respect to a schema where only position is optimized.

Sanz *et al.* [40] use GAs for improving the performance of fuzzy rule-based classification systems by adapting the fuzziness of (triangular) fuzzy sets as measured by the length of their support. In this work, interval-valued fuzzy sets are used to represent linguistic terms, and inference depends on lower and upper triangular fuzzy sets that may not satisfy the requirement of SFPs. Also, the use of triangular fuzzy sets only (though of type-2) does not assure that all possible fuzziness degrees are exploited to find the final fuzzy partitions. Nevertheless, experimental results show that accuracy can be improved by acting on the fuzziness of the fuzzy sets.

In many cases, triangular fuzzy sets are used to define the semantics of linguistic terms. In few works, however, trapezoidal fuzzy sets have been employed with the result of a greater flexibility. For example, Nguyen *et al.* [41] showed that the extension of hedge algebra semantics by trapezoidal fuzzy sets (in place of triangular) leads to

an average improvement of the resulting fuzzy rule-based classifiers in terms of both structural complexity and accuracy.

Evolutionary algorithms have been employed to optimize trapezoidal fuzzy sets for data-driven design of fuzzy partitions [42]. However, differently from triangular fuzzy sets, special care must be paid on the way fuzzy sets' parameters are fixed: without proper constraints, it is not guaranteed that the resulting partition satisfies the requirements of a SFP.

The aforementioned research works, alongside subsequent developments, stimulated a research question concerning the evaluation of the influence of the *sole* fuzziness in the data-driven optimization of a fuzzy partition, given a certain type of fuzzy rule-based system. To conduct this study, an optimization framework is needed with some key features:

- i. the optimization process must operate in the space of fuzziness only, so as to avoid the interference due to changing the position of fuzzy sets. To this purpose, a proper formalization of fuzziness is required;
- ii. the optimization process should potentially span all possible values of fuzziness for all the fuzzy sets involved in the fuzzy partitions that constitute the database of a fuzzy rule-based system;
- iii. the optimization process must not consider configurations of fuzziness that violate the requirements of SFPs.

The following sections are devoted to illustrate the proposed approach for optimizing the fuzziness in the way declared above. The main assumptions in this work are

1. we consider fuzzy rule-based classifiers only, with different defuzzification schemes and no rule weights involved. This is the simplest structure of rules that can be useful to highlight the influence of fuzziness in the final quality of the model;
2. we use trapezoidal fuzzy sets because they are more flexible than triangular fuzzy sets and provide a wider search space in terms of fuzziness. Additionally, fuzziness of trapezoidal fuzzy sets can be easily formalized in terms of the slopes of their oblique slides.

As a final note of caution, we underline how our proposal does not concern a new modeling technique (which would be improper since fuzzy sets are not modified in terms of position), rather it is focused on the analysis of the influence of fuzziness in the performance of the selected models.

3. CUT-BASED REPRESENTATION OF SFPs

With the purpose of separating position and fuzziness of the trapezoidal fuzzy sets involved in a fuzzy partition, we resort to a specific representation that is focused on the crossing points between adjacent fuzzy sets, which are henceforth called *cuts*.

Let $U = [m_U, M_U] \subseteq \mathbb{R}$ be a Universe of Discourse (UoD). A SFP can be composed on U by a sequence of normal and convex

fuzzy sets A_1, A_2, \dots, A_{n+1} if their membership functions μ_{A_i} are such that:

$$\forall x \in U : \sum_{i=1}^{n+1} \mu_{A_i}(x) = 1$$

For $i = 1, \dots, n$, we assume that the fuzzy sets involved in any couple (A_i, A_{i+1}) intersect in such a way that there exists a single point $t_i \in U$ such that $\mu_{A_i}(t_i) = \mu_{A_{i+1}}(t_i) = 0.5$, due to the particular arrangement of the SFP. In other words, a sequence of points t_1, \dots, t_n , such that $t_i < t_{i+1}$, can be identified in U and any range $[t_i, t_{i+1}]$ ($i = 1, \dots, n-1$) defines the boundaries of the 0.5-cut related to the fuzzy set A_{i+1} (see Figure 1 as an illustrative example with $n = 4$). Such points are referred to as cuts and play a crucial role in the context of our study. In fact, we are going to elect the sequence of cuts as a constraint to arrange an optimal SFP: assigned the points t_1, \dots, t_n , we want to explore the way to build onto them a partition which could prove to produce better results when employed in a fuzzy inference system. For each fuzzy set in the SFP, this approach allows to decouple its position, which is identified by two consecutive cuts, from its fuzziness, which stands as a degree of freedom that is susceptible of optimization.

Different types of fuzzy sets can be involved into the realization of a SFP; among them, the trapezoidal fuzzy sets are characterized by a four-parameter function $T[a, b, c, d]$ defined as

$$T[a, b, c, d](x) = \begin{cases} \frac{x-a}{b-a}, & x \in]a, b[\\ 1 & x \in [b, c] \\ \frac{x-d}{c-d}, & x \in]c, d[\\ 0 & x \leq a \vee x \geq d \end{cases}$$

with the parameters a, b, c, d subjected to the condition:

$$a \leq b \leq c \leq d \quad (1)$$

which ensures the well formedness of the fuzzy sets. Obviously, a trapezoidal fuzzy set assumes a triangular shape if the particular case $b = c$ occurs.

Indeed, the choice of the fuzzy set typology is significant in our research. Since we are dealing with the problem to design SFPs constrained by cuts, it is worth recalling that the employment of triangular fuzzy sets is a limiting choice, which would lead to the impossibility to produce triangular SFPs when some particular sequences of cuts are assigned. The adoption of trapezoidal fuzzy sets, instead, is flexible enough to guarantee the realization of well-formed SFPs whatever sequence of cuts is assigned, as demonstrated in a previous work of ours [30].

The fuzzy sets involved in a trapezoidal SFP must be such that

$$\begin{cases} a_1 = b_1 = m_U, \\ a_{i+1} = c_i, & (i = 1, \dots, n) \\ b_{i+1} = d_i, & (i = 1, \dots, n) \\ c_{n+1} = d_{n+1} = M_U \end{cases} \quad (2)$$

it can be noticed how the sequence of fuzzy sets is confined at the extremes by rectangular trapezoids leaning against the limits m_U and M_U of U .

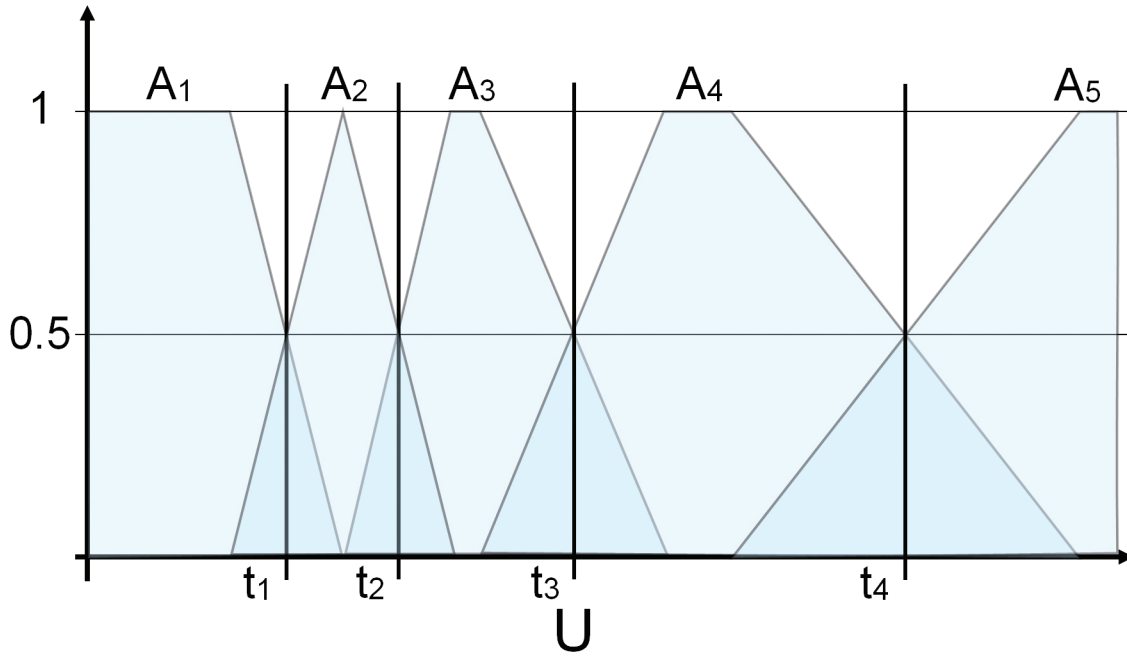


Figure 1 | A strong fuzzy partition (SFP) over the Universe of Discourse composed by 5 fuzzy sets whose positions are determined by the arrangement of 4 cuts.

To tailor a trapezoidal SFP to an assigned sequence of cuts t_1, \dots, t_n , an additional couple of conditions (which are of course equivalent due to (2)) must hold:

$$\begin{cases} b_{i+1} = 2t_i - a_{i+1} \\ d_i = 2t_i - c_i \end{cases} \quad (3)$$

The conditions expressed in (3) allow the cut t_i to be in the middle point between two consecutive vertices of a trapezoidal fuzzy set.

Several methods may be adopted to build a trapezoidal SFP from cuts. The Constant Slope (CS) method [30] is a very simple and intuitive one, which can be applied in full compliance with the aforementioned conditions (2–3). According to the CS method, a common slope is set for all the trapezoidal fuzzy sets involved in the resulting SFP. This specific slope is the one¹ associated to the sides of the triangular fuzzy set centered on the middle point between the nearest cuts in the assigned sequence. Figure 2 illustrates an application of the CS method: the sequence of cuts t_1, t_2, t_3 is assigned to produce a partition involving 4 fuzzy sets. Firstly, the minimum distance Δ_{min} among cuts is identified

$$\Delta_{min} = \min\{(t_{i+1} - t_i) | i = 0, \dots, 3\}$$

where $m_U = t_0$ and $M_U = t_4$ have been imposed to integrate the sequence of cuts (this practice is going to be replied in the rest of the present work). Then, a triangular fuzzy set (labeled as A_2) is built onto Δ_{min} and its specific slopes on the left and the right sides are imposed to construct the remaining trapezoidal fuzzy sets composing the resulting SFP.

¹Actually, two opposite slopes are involved (related to the ascending and the descending sides of the fuzzy set, respectively) which are equal in magnitude.

Other methods can be adopted to derive trapezoidal SFPs from cuts: each of them is supposed to provide different results [30]. All of the partitions are characterized by a number of trapezoidal fuzzy sets with fixed positions (due to the prearranged sequence of cuts) and exhibit dissimilar fuzziness as determined by the different slope configurations of the oblique slides. However, a single method is able to produce a single partition of fuzzy sets, against a number of possibilities which is infinite. Hence the necessity to devise a way for fully exploring this space of possibilities to properly investigate how fuzziness affects the performance of such SFPs when they are applied in some contexts of application.

4. PARTICLE SWARM OPTIMIZATION AS A TOOL TO OPTIMIZE FUZZINESS

Particle swarm optimization (PSO) is a computational method devised to perform stochastic optimization [43]. Originally introduced in 1995 [44], it soon started attracting interest from researchers and underwent further analysis, variations, and development [45,46,47,48].

PSO emulates social behavior with special references to such organized groups as insect swarms, fish schools, and bird flocks. All of them are composed by a large number of individuals which all together interact and coordinate their movements to contribute to the global repositioning of the group. In this way, the whole set of individuals is able to reach some goal of common interest. Drawing inspiration from this natural behavior, PSO incorporates a number of entities moving through a search space to find a (nearly-)optimal solution for a given problem. Movement and velocity are key concepts, so the involved entities are called “particles,” a term which best suits these notions. Just like in a flock or a swarm, the particles interact with one other and, at the same time, they learn from their

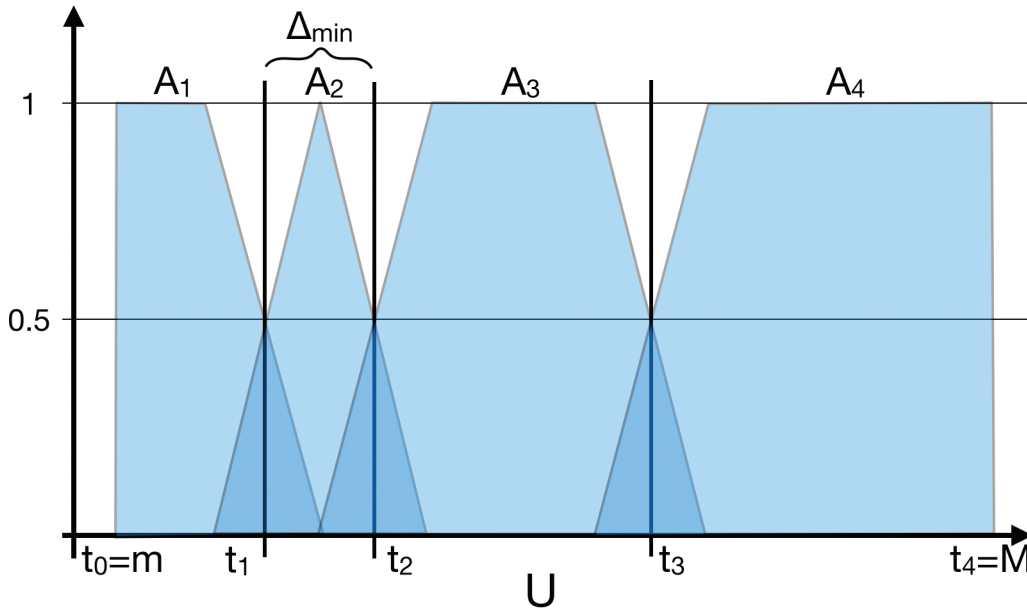


Figure 2 | A trapezoidal strong fuzzy partition (SFP) tailored on the basis of 3 assigned cuts through the application of the constant slope (CS) method.

own experience. To keep track of advancements, PSO evaluates the particles' positions through an objective function (which must be minimized) and then moves the population at each successive iteration of the process. The movement of each particle through the search space is determined by combining information about its current and best position, the best positions reached by one or more other particles in the swarm, and some random perturbations. As a global result, the population gradually moves toward better regions of the search space, reaching a final position which can be regarded as the solution of the problem at hand.

Given an objective function $f : E \rightarrow \mathbb{R}$ to be minimized, a formal description of the R -dimensional search space E to be explored by PSO can be expressed as follows:

$$E = [l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_R, u_R] \quad (4)$$

being $[l_r, u_r] \subset \mathbb{R}$ for $r = 1, 2, \dots, R$. Let S be the number of particles composing the swarm which is going to explore E : at time τ , each of them is associated to three R -dimensional vectors:

- $\mathbf{x}_i(\tau)$, a position vector encoding the current location of the particle;
- $\mathbf{v}_i(\tau)$, a velocity vector which previously contributed to determine the current position vector of the particle;
- $\mathbf{p}_i(\tau)$, a memory vector encoding the best position reached so far,

for $i = 1, \dots, S$. A further piece of information regards the particle neighborhood (possibly, the entire swarm) and it is encoded in the R -dimensional vector:

- $\mathbf{p}_g(\tau)$, a position vector encoding the best position globally reached so far.

By the term “best” position, we refer to the evaluation of a position vector performed through the objective function, which is purposely designed depending on the problem at hand.

The PSO is organized as an iterative process: at each step, the position and velocity vectors of each particle are updated as follows:²

$$\begin{cases} \mathbf{v}_i(\tau + 1) = \omega \mathbf{v}_i(\tau) + \mathbf{u}(0, \phi_1) \circ (\mathbf{p}_i(\tau) - \mathbf{x}_i(\tau)) \\ \quad \quad \quad + \mathbf{u}(0, \phi_2) \circ (\mathbf{p}_g(\tau) - \mathbf{x}_i(\tau)) \\ \mathbf{x}_i(\tau + 1) = \mathbf{x}_i(\tau) + \mathbf{v}_i(\tau + 1) \end{cases} \quad (5)$$

where \circ is the component-wise multiplication; ω is a positive value (the *inertia* weight) which can be regarded as the fluidity of the swarm motion; $\mathbf{u}(0, \phi_i)$ are R -dimensional vectors composed by random numbers uniformly distributed in $[0, \phi_i]$; ϕ_1, ϕ_2 are positive values (the *cognitive* and *social* coefficients) which can be regarded as the attraction of the i -th particle towards the best of its positions and the global best position, respectively.

To some degree, the PSO working machinery resembles another nature-inspired algorithm, i.e., the GA, which is also a stochastic population-based optimization process. Both particles and chromosomes similarly act as candidate solutions and the objective function in PSO clearly recalls the fitness function of GAs. However, some differences between these methods can be highlighted [49]. PSO does not implement selection and lacks genetic operators such as crossover and mutation (even if a kind of crossover is represented by the combined information of particle and neighborhood best positions, useful to evaluate acceleration). Also, PSO is intrinsically directional in its process, while the GA mutation is omnidirectional in its nature. All in all, PSO is able to produce satisfactory results in terms of optimization while showing the benefit of an easier implementation and a reduced number of parameters to adjust

²As previously asserted, some other formulations for updating the PSO vectors have been proposed in literature, but we can refer to this version for our purposes.

when compared with GA. It applies, therefore, as a suitable candidate to carry on the exploration of the search space we are going to investigate in the present work.

The scenario under study is such that the SFPs are meant to be exploited to build up a fuzzy rule-based inference system from data. Therefore, we deal with the following setting:

- (a) the cuts are supposed to be imposed by the specific context and derived from the analysis of the dataset at hand (by means of a particular clustering process);
- (b) the objective function steering the PSO process is represented by the average accuracy obtained while evaluating the rule base corresponding to the derived SFP;
- (c) the particles' position in the search space must be related to the shape of trapezoidal fuzzy sets.

It can be observed how, according to (a), the locations of the involved trapezoidal fuzzy sets are fixed in our setting.

We already applied PSO to design trapezoidal SFPs in a preliminary study of ours [31]. In particular, concerning point (c), we put a direct correspondence between the particles and the edges of a trapezoidal fuzzy set. More precisely, given a sequence of cuts t_1, t_2, \dots, t_n , recalling that a SFP is fully characterized by the sequence of $4(n+1)$ parameters

$$a_1, b_1, \dots, \underbrace{c_i, d_i, a_{i+1}, b_{i+1}}, \dots, c_{n+1}, d_{n+1} \quad (6)$$

we observe that each sub-sequence $c_i, d_i, a_{i+1}, b_{i+1}$ can be reduced to c_i . In fact, from (2) and (3) we get $a_{i+1} = c_i$ and $b_{i+1} = d_i = 2t_i - c_i$. From being also $a_1 = b_1 = m_U$ and $c_{n+1} = d_{n+1} = M_U$, we conclude that the sequence (6) can be fully recovered by the sequence

$$c_1, c_2, \dots, c_n \quad (7)$$

which includes the only free parameters that can be modified in order to get different SFPs, provided that the bounds m_U, M_U and the cuts t_1, \dots, t_n are preserved.

Moving from those assumptions, it is straightforward to consider an n -dimensional search space³ where the particles can shift their position vectors which ultimately correspond to sequences of trapezoid edges as in (7). From (4) we know that the search space is composed by bounded intervals and in our past study we related them to the ranges where each trapezoid edge $c_i \in [l_i, u_i]$ must find place.

³Up to now, we considered the SFP which can be drawn on a single UoD. When dealing with real-world problems, the related datasets usually involve several dimensions (say, D), thus implying that the same reasoning scheme must be replied for all the involved UoDs. In this sense, a sequence of cuts must be provided for each dimension, $(t_1, t_2, \dots, t_{n_1}), \dots, (t_1, t_2, \dots, t_{n_D})$, which in turn requires the PSO to look for a number of D sequences of free parameters $(c_1, c_2, \dots, c_{n_1}), \dots, (c_1, c_2, \dots, c_{n_D})$, thus enlarging the dimension of the search space up to the value $n_1 + n_2 + \dots + n_D$. For the sake of simplicity, we take for granted such a clarification and we keep it implicit, while continuing to refer in the following to the case of a single UoD.

Since the sequence of cuts $m_U = t_0, t_1, t_2, \dots, t_n, t_{n+1} = M_U$ is assigned, it can be easily inferred that

$$c_i \in [l_i, u_i] \subseteq [t_{i-1}, t_i] \quad \text{for } i = 1, \dots, n+1,$$

and special attention must be paid to correctly specify the subsets of the intervals defined by the cuts where every c_i should be placed. In fact, their arbitrary positioning inside $[t_{i-1}, t_i]$ could compromise the well-formedness of the resulting trapezoidal fuzzy sets, which is to be preserved instead.

Therefore, we developed a couple of strategies to constrain the placement of the c_i edges inside each range, basically founded on a geometrical analysis of the SFP design problem. The strategies we adopted were termed Leftmost Slope Constraint and Constant Slope Constraint (we address the interested reader to Ref. [31] for further details about them): they have been exploited to properly define the bounded intervals composing the search space which has been then explored by the PSO process being assured about the well-formedness of all the trapezoidal SFPs.

The results we obtained were encouraging, however we were conscious that the imposed constraints reduced the search space as a side effect. In other words, during the PSO process the particles could not explore all the infinite possible solutions of the problem at hand, but a subset of them (which is different depending on the specific strategy adopted to constrain the ranges $[l_i, u_i]$). This is the reason why we developed an alternative strategy to relate the particles' positions and the edges of the trapezoidal fuzzy sets, as we are going to discuss in the following section.

5. A NEW DEFINITION OF THE PSO SEARCH SPACE

The preliminary tests showed that PSO stands as a suitable tool to support the design of SFPs based on cuts. However, to fully appreciate the effectiveness of this approach, a thorough exploration of the search space must be performed. In this way, we can evaluate how much the PSO algorithm is able to fine-tune the fuzziness of the fuzzy sets involved in an inference system whose realization is constrained by some assigned cuts. In this section, therefore, we introduce a novel formalization of the particles involved in the PSO process and we demonstrate that it is instrumental in shaping a search space including *all* (and *only*) the trapezoidal SFPs possibly standing on a UoD.

To this aim, we alter the meaning of the particle positions in the search space: instead of being regarded as the set of edges c_1, \dots, c_n of the trapezoidal fuzzy sets, they are simply interpreted as a set of values $\alpha_1, \dots, \alpha_n \in [0, 1]$. By doing so, the bounded intervals composing the search space (see equation (4)) assume a fixed form:

$$[l_i, u_i] = [0, 1] \quad \text{for } i = 1, \dots, n$$

To derive the edges of the fuzzy sets, we proceed as follows. Assigned a sequence of cuts $m_U = t_0, t_1, t_2, \dots, t_n, t_{n+1} = M_U$ over a UoD $U = [m_U, M_U]$, we define:

$$a_1 = b_1 = t_0 \quad (8a)$$

$$a_{i+1} = c_i = (t_i - s_i)\alpha_i + s_i \quad (8b)$$

$$b_{i+1} = 2t_i - a_{i+1} \quad (8c)$$

$$d_i = 2t_i - c_i \quad (8d)$$

$$c_{n+1} = d_{n+1} = t_{n+1} \quad (8e)$$

It can be observed how all the points a_i, b_i, c_i, d_i framing the fuzzy sets are defined for $i = 1, \dots, n+1$. Particularly, the sequence c_1, \dots, c_n is evaluated in terms of $\alpha_1, \dots, \alpha_n$, i.e. the previously introduced values provided by the PSO search. Also, they are defined in terms of s_1, \dots, s_n , being

$$s_i = \max[(2t_i - t_{i+1}), b_i] \quad \text{for } i = 1, \dots, n \quad (9)$$

Figure 3 provides an intuitive understanding of the s_i placement with respect to the cuts positions. The configuration depicted in Figure 3(a) is such that $b_i < (2t_i - t_{i+1})$: in this case a safe area to set the c_i point is determined (represented by the green region in figure), ranging from $(2t_i - t_{i+1})$ to t_i . Setting c_i outside from this area would disrupt the SFP design, as shown by the red dashed trapezoidal fuzzy set in figure which intersects the cut t_{i+1} . The configuration depicted in Figure 3(b) is such that $(2t_i - t_{i+1}) < b_i$: in this case a safe area to set the c_i point is ranging from b_i to t_i . Setting c_i outside from the safe area would disrupt the well-formedness of the trapezoidal fuzzy set.

Moving from the above assumptions, we are going to demonstrate that, as the values $\alpha_1, \dots, \alpha_n$ vary in $[0, 1]$, it is possible to explore the whole space of the trapezoidal SFPs which can be drawn on a UoD. The proof is split into two theorems to prove that each sequence $\alpha_1, \dots, \alpha_n$ allows to derive: (i) only well-formed trapezoidal SFPs; (ii) all the obtainable well-formed trapezoidal SFPs.

Firstly we must consider the following lemmas:

Lemma 1 Let $U = [m_U, M_U] \subseteq \mathbb{R}$ be a UoD and let $t_1, t_2, \dots, t_n \in U$, with $t_i < t_{i+1}$. Let $t_0 = m_U, t_{n+1} = M_U$ and let $\alpha_1, \alpha_2, \dots, \alpha_n \in [0, 1]$. For $i = 1, \dots, n$, given the position (8b), the following inequality holds:

$$c_i \geq 2t_i - t_{i+1} \quad (10)$$

Proof. From being $c_i = (t_i - s_i)\alpha_i + s_i = t_i\alpha_i + s_i(1 - \alpha_i)$, we can evaluate c_i by distinguishing two alternative cases:

$$s_i = \begin{cases} 2t_i - t_{i+1} & \text{for } b_i < 2t_i - t_{i+1} \\ b_i & \text{for } 2t_i - t_{i+1} \leq b_i \end{cases} \quad (11a)$$

$$(11b)$$

From (11a) we obtain

$$\begin{aligned} c_i &= t_i\alpha_i + (2t_i - t_{i+1})(1 - \alpha_i) \\ &= (2t_i - t_{i+1}) + \alpha_i(t_{i+1} - t_i) \\ &\Rightarrow c_i \geq 2t_i - t_{i+1} \end{aligned} \quad (12)$$

On the other hand, from (11b) we obtain

$$\begin{aligned} c_i &= t_i\alpha_i + b_i(1 - \alpha_i) \geq t_i\alpha_i + (2t_i - t_{i+1})(1 - \alpha_i) \\ &\Rightarrow c_i \geq 2t_i - t_{i+1} \end{aligned}$$

Lemma 2 Given the same premises of Lemma 1, for $i = 1, \dots, n$ the following inequality holds:

$$s_i \leq t_i \quad (13)$$

Proof. For $i = 1$ we should prove that $s_1 \leq t_1$ while considering the possible values of s_1 :

$$s_1 = \begin{cases} 2t_1 - t_2 = t_1 + (t_1 - t_2) \\ b_1 = t_0 \end{cases} \Rightarrow s_1 \leq t_1$$

For $i = 2, \dots, n$, we can prove that $s_i \leq t_i$ by *reductio ad absurdum* while considering the possible values of s_i . Whenever $(2t_i - t_{i+1}) > b_i$ we get $s_i = 2t_i - t_{i+1}$ and, by contradiction with (13), the following inequality would result:

$$2t_i - t_{i+1} > t_i \Rightarrow t_i > t_{i+1}$$

which is absurd since we posed that $t_1, t_2, \dots, t_n \in U$, with $t_i < t_{i+1}$.

Whenever $(2t_i - t_{i+1}) < b_i$ we get $s_i = b_i = 2t_{i-1} - a_i = 2t_{i-1} - c_{i-1}$ and, by contradiction with (13), the following inequality would result:

$$2t_{i-1} - c_{i-1} > t_i \Rightarrow c_{i-1} < 2t_{i-1} - t_i \quad (14)$$

which is absurd due to Lemma 1.

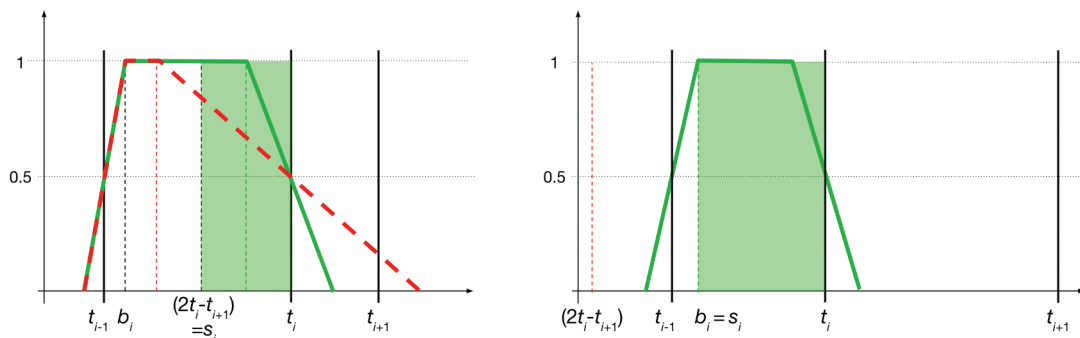


Figure 3 | Placement of $s_i = \max[(2t_i - t_{i+1}), b_i]$. According to the illustrated cuts configurations, $s_i = (2t_i - t_{i+1})$ in (a); $s_i = b_i$ in (b).

Theorem 3 Let $U = [m_U, M_U] \subseteq \mathbb{R}$ be a UoD and let $t_1, t_2, \dots, t_n \in U$, with $t_i < t_{i+1}$. Let $t_0 = m_U, t_{n+1} = M_U$ and let $\alpha_1, \alpha_2, \dots, \alpha_n \in [0, 1]$. For $i = 1, \dots, n$, if the positions (8a)-(8e) are given, then by varying the values of $(\alpha_1, \alpha_2, \dots, \alpha_n)$ inside the hypercube $[0, 1]^n$, the (infinite) partitions obtained on the UoD are trapezoidal SFPs tailored on t_1, \dots, t_n (according to (3)).

Proof. To construct a well-formed trapezoidal SFP the following relationship must hold:

$$a_i \leq t_{i-1} \leq b_i \leq c_i \leq t_i \leq d_i \quad \text{for } i = 1, \dots, n+1 \quad (15)$$

as well as the equalities expressed in (8b)-(8d).

From Lemma 2 we argue that $c_i \in [s_i, t_i]$ for $i = 1, \dots, n$. From being $b_i \leq s_i$, we are certain about the relationship $b_i \leq c_i \leq t_i$ for $i = 1, \dots, n$. Therefore, from (8d), we get also $c_i \leq t_i \leq d_i$ for $i = 1, \dots, n$ (the case related to $i = n+1$ refers to the rightmost rectangular trapezoid, where c_i, d_i , and t_i are collapsing to the same point M_U). Finally, from (8c), we get $a_{i+1} \leq t_i \leq b_{i+1}$ for $i = 2, \dots, n$ (the case related to $i = 1$ refers to the leftmost rectangular trapezoid, where a_i, b_i , and t_0 are collapsing to the same point m_U).

In summary, the relationship (15) is proven and we can argue that, however set the values of $(\alpha_1, \alpha_2, \dots, \alpha_n)$ in $[0, 1]^n$, the resulting partition of the UoD is a well-formed trapezoidal SFP.

Theorem 4 Let $U = [m_U, M_U] \subseteq \mathbb{R}$ be a UoD and let $t_1, t_2, \dots, t_n \in U$, with $t_i < t_{i+1}$. Let $t_0 = m_U, t_{n+1} = M_U$. For $i = 1, \dots, n$, all the (infinite) well-formed trapezoidal SFPs on U tailored on t_1, \dots, t_n (according to (3)) can be obtained by varying the values of $(\alpha_1, \alpha_2, \dots, \alpha_n)$ inside the hypercube $[0, 1]^n$.

Proof. If the values of $(\alpha_1, \dots, \alpha_n)$ are to be involved into the construction of the SFPs, then all the vertexes framing the trapezoidal fuzzy sets must be such that $c_i \in [s_i, t_i]$. This is due to the definition expressed in (8b) when we consider that $s_i \leq t_i$ (from Lemma 2) and $\alpha_i \in [0, 1]$ for $i = 1, \dots, n$. By contradiction of the thesis, therefore, there should be a trapezoidal SFP such that

$$c_i < s_i \quad \text{or} \quad (16a)$$

$$c_i > t_i \quad (16b)$$

However, the inequality in (16b) contrasts with (15), thus ruining the well-formedness of the resulting trapezoidal SFP. Concerning the inequality in (16a), from being $s_i = \max[(2t_i - t_{i+1}), b_i]$, it translates into

$$c_i < b_i \quad \text{or} \quad (17a)$$

$$c_i < 2t_i - t_{i+1} \quad (17b)$$

The inequality in (17a) again contrasts with (15). The inequality in (17b) implies

$$2t_i - c_i = d_i > t_{i+1}$$

which would ruin the well-formedness of the resulting trapezoidal SFP. In summary, neither (16a) nor (16b) would lead to the definition of well-formed trapezoidal SFPs, thus proving that all of them can be only obtained in compliance with the theorem hypotheses.

We have shown that by varying the values of $(\alpha_1, \alpha_2, \dots, \alpha_n)$ inside the hypercube $[0, 1]^n$ it is possible to derive all and only the trapezoidal SFPs which can be designed on the UoD U . The alternative design strategy represented by the positions (8a-8e), therefore, can be adopted to exhaustively explore the SFP search space, exploiting the PSO algorithm to generate optimal sequences of the parameters α_i .

It can be observed that the parameters c_i do not linearly depend on the parameters a_j ($j < i$) used to determine them. In principle, this means that the SFPs are not uniformly distributed in the search space to be explored. That is true at the beginning of the PSO process, even if the parameters c_i are generated by a uniform random sequence $(\alpha_1, \alpha_2, \dots, \alpha_n)$. To attenuate the effects of the lack of uniformity, PSO will be set up with a large number of particles and a large number of iterations to ensure that low-probability configurations have the chance of being explored if their fitness is high.

6. EXPERIMENTAL RESULTS

Trapezoidal SFPs may be useful to design the structure of inference systems based on fuzzy rules. We already mentioned how this idea has been injected in the very definition of the previously discussed PSO algorithm, whose objective function is represented by the accuracy evaluation of the resulting fuzzy system applied on some dataset. The cuts themselves can be interpreted as constraints intrinsic to the problem at hand, which can be extracted from the dataset analysis. In our work, we are not interested in tackling some specific problem aiming at designing an optimal fuzzy inference engine to solve that particular task at its best. Our goal is to test the suitability of the developed PSO strategy in fine-tuning the design of trapezoidal SFPs based on cuts, instead. To this aim, we refer to a number of datasets related to classification problems: we are going to use them to evaluate the performance evolution of some fuzzy rule-based systems while optimizing the fuzziness of the fuzzy sets involved in the underlying fuzzy partitions.

We set up an array of datasets, including both synthetic and real benchmark data. The first are described in Table 1: they are all bi-dimensional datasets purposely designed to include a diverse number of samples and classes. The benchmark datasets are described in Table 2: they come from publicly accessible repositories and can be retrieved online. [50,51,52]. Again, they represent a variety of cases in terms of samples, features, and classes.

The cuts needed to trigger the design process can be acquired directly from data by means of a clustering process. In particular,

Table 1 | Description of the synthetic datasets involved in the experimental session.

Dataset	# Samples	# Features	# Classes
SD1	200	2	3
SD2	400	2	3
SD3	400	2	3
SD4	300	2	3
SD5	600	2	3
SD6	500	2	2
SD7	300	2	2
SD8	600	2	3

we employed DC*, an algorithm which takes its name from Double Clustering with A*, i.e., a mechanism oriented to extract interpretable information from data [53]. That is accomplished by going through a two-step process which firstly clusters the data around some detected prototypes and then projects the prototypes themselves along each feature axis to allow a further clustering performed by the A* algorithm. By doing so it is possible to derive a partition over each dimension: this represents the setting where the SFPs can be grounded, thus enabling the design of a fuzzy rule-based system incorporating the fuzzy sets which compose the obtained fuzzy partitions. This process is carried out by the DC* algorithm intervening on every dimension at the same time: some of them could be unaffected by the partition construction, thus implicitly reducing the dimensionality of the problem at hand. The interested reader can find a thorough description of DC* in Ref. 53.

The cuts play a pivotal role in the previously described algorithm since they represent the mid-points placed on each axis between a couple of projections referring to prototypes of different classes. As such, they contribute to the definition of the final configuration of partitions: a subset of cuts is identified through the A* search to split the space of the problem into sub-spaces containing homogeneous prototypes.

For the purposes related to the present work we are going to exploit DC* not to benefit of its main results (namely, the definition of interpretable granules of information to be embedded into an inference engine in form of fuzzy rules), but to profit from its side products, i.e., the cuts derived from data. As previously asserted, they are necessary to start up the trapezoidal SFP optimization process. In this sense, we are not concerned with the attainment of the best clustering results which DC* may provide. For that reason, in some cases we contented with sub-optimal cut configurations. That can be noticed by referring to Figure 4 where the datasets SD1-SD8 are represented together with the cuts obtained at the end of the application of the DC* algorithm (being bi-dimensional, the synthetic

datasets lend themselves to a handy graphical illustration). In a similar way, a configuration of cuts has been obtained through DC* for each of the benchmark datasets listed in Table 2.

Once the cuts are assigned, an infinite number of partitions can be drawn on the space of data. To assess the capability of the PSO algorithm as a fine-tuning tool of trapezoidal SFPs, we firstly set a landmark method to be employed in the following for the sake of comparison. That is represented by the CS algorithm described in Section 3, which is one of the default methods adopted by DC* to finally produce the trapezoidal fuzzy sets composing the fuzzy inference system resulting from the data clustering process. We launched the PSO algorithm following the scheme reported in Equation (5), posing $\omega = 0.7298$ and $\phi_1 = \phi_2 = 1.49618$: these parameter values are consistent with some suggestions reported in literature [54,55,56]. Furthermore, we set 1000 particles per swarm and 1000 iterations, with an early stopping after 200 iterations without advancement of the objective function. It should be recalled that the particles are free to move in the search space E and the trapezoidal fuzzy set corresponding to each particle position is derived by applying equations (8a–8e), as reported in Section 5.

A twofold inference strategy has been set up to assess the performance of the fuzzy systems built up on the SFPs obtained through the CS method. On the one hand, a winner-take-all strategy (labeled as “MAX” method) assigns to each sample in the dataset the class related to the fuzzy rule which activates at the highest degree. On the other hand, an alternative strategy (labeled as “SUM” method) distinguishes the groups of fuzzy rules providing the same classes and assigns to each sample in the dataset the class related to the group producing the highest cumulative strength. Similarly, the objective function driving the search of the PSO algorithm embeds both the MAX and the SUM methods, so that the fuzzy systems corresponding to each particle position produce two kinds of inference results too. In this way, a congruent comparison can be performed and the fine-tuning capabilities of the PSO can be assessed for both the inference methods.

Tables 3 and 4 illustrate a comparison (in terms of accuracy performance) of the fuzzy rule-based systems resulting from the partitions obtained by the CS method and the PSO full search (FS). The comparison refers to the classification of the synthetic data applying the MAX and SUM inference methods, respectively. In the same way, Tables 5 and 6 illustrate an analogous comparison referred to the classification of the benchmark data.

As a general remark, we can observe the suitability of the PSO algorithm as a tool for fine-tuning the derived SFPs based on cuts. In

Table 2 | Description of the benchmark datasets involved in the experimental session. (#S = number of samples, #F = number of features, #C = number of classes.)

Dataset	ID	#S	#F	#C
Appendicitis	Ap	106	7	2
Balance	Bl	625	4	3
Banana	Bn	5300	2	2
Beer Styles	BS	400	3	8
Bupa	Bu	345	6	2
Cardiotocography	CTG	2126	21	3
Hayes-Roth	Hy	160	4	3
Ionosphere	Ion	351	33	2
Iris	Ir	150	4	3
Monk-2	Mo	432	6	2
Newthyroid	Nth	215	5	3
Page-Blocks	PB	5472	10	5
Phoneme	Ph	5404	5	2
Pima	Pi	768	8	2
Saheart	Sh	462	9	2
Sonar	So	208	60	2
Thyroid	Thy	7200	21	3
Vertebral 2	V2	310	6	2
Vertebral 3	V3	310	6	3
Wine	Wi	178	13	3

Table 3 | Fuzzy classification of synthetic data — Fine-tuning operated by the PSO search with respect to CS using the MAX inference method.

Dataset	Accuracy Performance (%)		Accuracy gain (%)
	CS	FS	
SD1	82.00	82.00	0.00
SD2	96.25	96.25	0.00
SD3	81.75	95.00	13.25
SD4	55.33	60.00	4.67
SD5	88.17	88.50	0.33
SD6	68.00	70.40	2.40
SD7	66.67	66.67	0.00
SD8	58.67	58.67	0.00

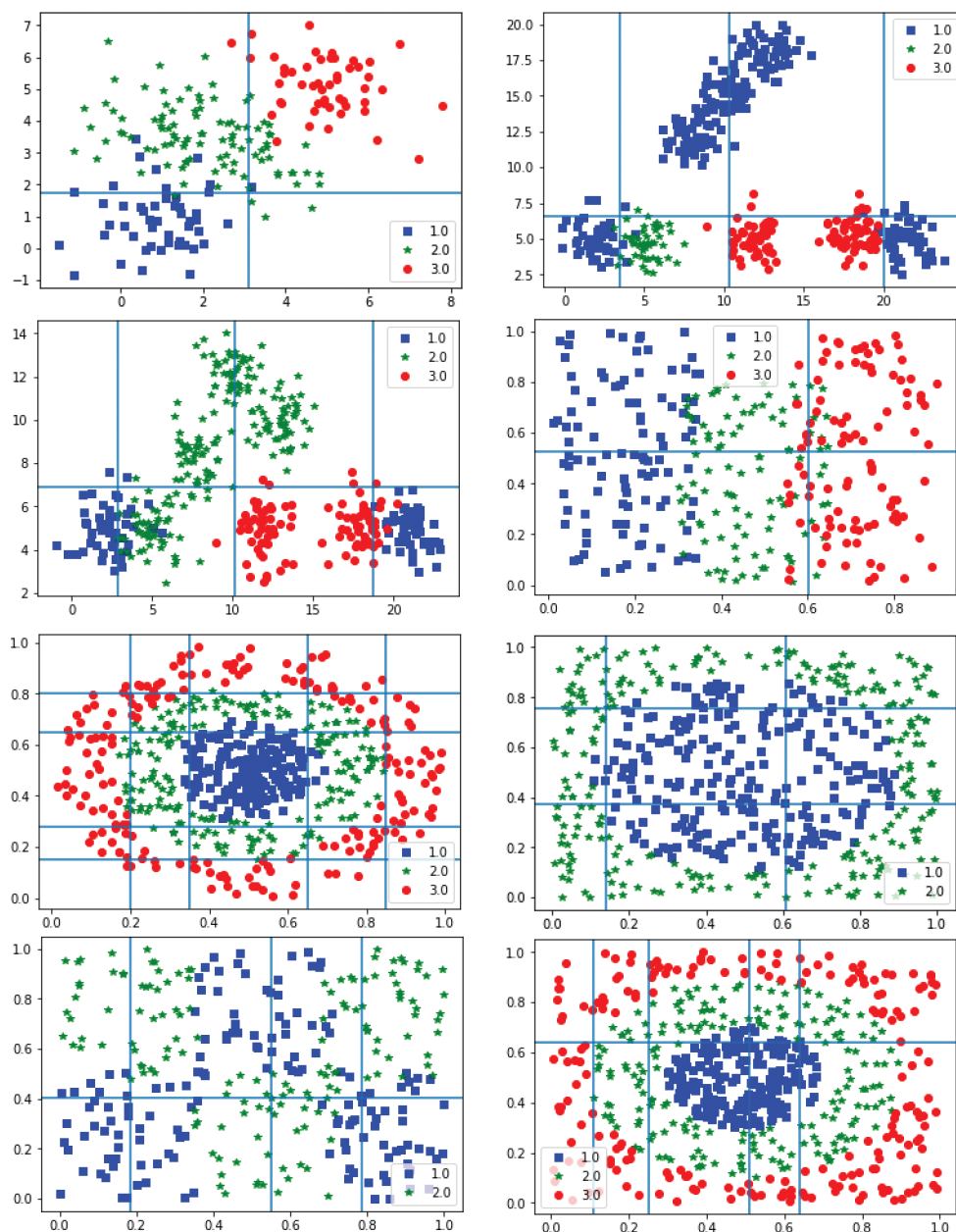


Figure 4 | Graphical representation of the synthetic datasets adopted for the experimental session. The figures include the cuts produced by DC*.

fact, in several cases its application contributes to modify the fuzziness of the involved fuzzy sets in such a way that the performance of the resulting classifiers is improved when compared to the results related to the baseline CS method. For some datasets the accuracy gain appears to be quite considerable. Also, none of the experiments determined a decay in the classification accuracy. It should be stressed, however, that the goal of our experiments concerned the analysis of the fuzziness influence on the final performance, instead of the definition of the best single model to be employed as an optimal classifier. In other words, we did not perform model selection; that is the reason why we did not apply any scheme of cross-validation, as it is usually the case in some other common experimental sessions whose scope is quite different from ours.

As a further remark, if we consider the results reported in the tables, it is straightforward to argue that the application of the PSO algorithm is much more profitable when fuzzy classifiers equipped with the SUM inference method are considered. This allows some interesting observations which can be brought to the attention of the research community dwelling on fuzzy system design. On the one hand, the performance of the MAX inference method exhibited a certain stiffness with respect to the actual orientation of the oblique slides composing the trapezoidal fuzzy sets.

In this sense, the main contribution to the final performance of the fuzzy classifiers comes from the position of the fuzzy partitions (which in our experimental session has been fixed by the cuts obtained from data through the application of the DC* algorithm).

Table 4 | Fuzzy classification of synthetic data — Fine-tuning operated by the PSO search with respect to CS using the SUM inference method.

Dataset	Accuracy Performance (%)		Accuracy gain (%)
	CS	FS	
SD1	82.00	82.00	0.00
SD2	86.75	96.25	9.50
SD3	92.50	92.75	0.25
SD4	55.33	60.00	4.67
SD5	87.67	91.33	3.67
SD6	69.00	77.40	8.40
SD7	61.33	67.67	6.33
SD8	58.17	61.67	3.50

Table 5 | Fuzzy classification of benchmark data — Fine-tuning operated by the PSO with respect to CS using the MAX inference method.

Dataset	Accuracy Performance (%)		Accuracy gain (%)
	CS	FS	
Ap	82.08	82.08	0.00
Bl	66.08	66.08	0.00
Bn	67.30	67.32	0.02
BS	95.00	95.50	0.50
Bu	61.45	62.32	0.87
CTG	75.82	75.82	0.00
Hy	44.38	44.38	0.00
Ion	65.24	65.24	0.00
Ir	88.00	90.00	2.00
Mo	36.11	36.11	0.00
Nth	92.56	92.56	0.00
PB	85.31	85.64	0.33
Ph	61.01	62.56	1.55
Pi	62.11	62.11	0.00
Sh	63.64	64.29	0.65
So	56.73	56.73	0.00
Thy	91.22	91.22	0.00
V2	56.13	56.13	0.00
V3	72.26	72.26	0.00
Wi	81.46	81.46	0.00

On the other hand, the performance of the SUM inference method proved to be much more sensitive to the fuzziness component, allowing a greater degree of variation in the optimization results while considering both the synthetic and the benchmark data.

As a consequence, to fully exploit the contribution coming from the fuzziness component, it makes sense to combine the adoption of the SUM inference method and trapezoidal fuzzy sets. As previously asserted, in fact, this kind of fuzzy sets lend itself to a major flexibility in terms of fuzziness, which cannot be attained with different set shapes (i.e., triangular fuzzy sets). It is worth noting that such observations are supported by an empirical analysis which has been thoroughly conducted during the application of both the inference methods. Indeed, as formally demonstrated in Section 5, the PSO algorithm has been applied to the problem at hand so as to exhaustively explore the search space composed by *all* the trapezoidal SFPs possibly standing on a UoD.

Finally, to explicitly illustrate the tuning process, Figures 5 and 6 depict the comparison of the trapezoidal SFPs derived for some datasets through the application of the baseline CS method and the PSO algorithm. Particularly, the figures illustrate the results concerning a selected subset of synthetic and benchmark datasets, respectively.

Table 6 | Fuzzy classification of benchmark data — Fine-tuning operated by the PSO with respect to CS using the SUM inference method.

Dataset	Accuracy Performance (%)		Accuracy gain (%)
	CS	FS	
Ap	86.79	86.79	0.00
Bl	66.08	66.08	0.00
Bn	65.34	68.81	3.47
BS	95.25	96.00	0.75
Bu	56.23	64.93	8.70
CTG	82.03	82.08	0.05
Hy	44.38	44.38	0.00
Ion	68.95	68.95	0.00
Ir	72.00	95.33	23.33
Mo	36.11	36.11	0.00
Nth2	83.26	92.56	9.30
PB	87.63	87.81	0.18
Ph	60.16	66.47	6.31
Pi	55.60	63.28	7.68
Sh	65.15	67.10	1.95
So	53.85	58.17	4.33
Thy	93.51	93.56	0.04
V2	49.68	57.42	7.74
V3	62.26	78.06	15.81
Wi	68.54	87.08	18.54

7. CONCLUSIONS

Designing fuzzy classifiers is a challenging endeavor involving a number of research issues. We deal with this matter by analyzing some convenient way to define the SFPs which underlie a fuzzy inference system. Particularly, we highlighted a couple of features that characterize the design of the fuzzy sets involved in a SFP, namely their position and fuzziness. While focusing on a specific shape among others, i.e., trapezoidal fuzzy sets, we investigated the contribution coming from fuzziness when the performance of fuzzy classifiers is evaluated. To this end, we set up an optimization method based on the PSO algorithm to fine-tune the slopes of trapezoidal sets.

We described a mechanism to model the fuzzy sets by adopting a particular PSO implementation: we formally demonstrate that our proposal enables an exhaustive exploration of the search space composed by *all* the possible SFPs traceable on a UoD once the position of fuzzy sets is fixed. In other words, we were able to provide a thorough assessment of the fuzziness contribution to the performance of fuzzy classifiers. In this sense, this work represents a completion of some previous research of ours, where the PSO algorithm has been adopted to provide a partial exploration of the above-described search space. Additionally, the experimental session allowed to bring into focus some interesting issues which can be of some relevance for the research community working on the design of fuzzy inference system. This kind of remarks are connected to a couple of inference methods which may be adopted to trigger the classification process of a fuzzy system: we termed them MAX and SUM methods.

As a hint for future research, we address the theoretical investigation of such inference methods as a major issue to better understand the semantics behind the working engine propelling a fuzzy classifier.

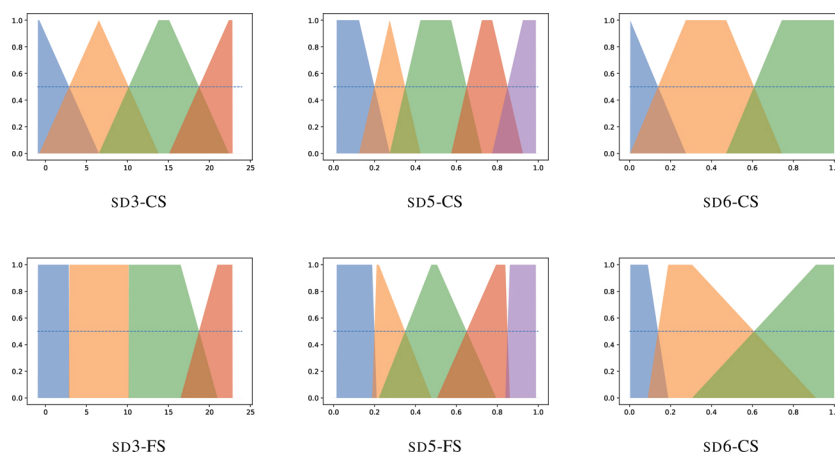


Figure 5 Comparison of the trapezoidal SFPs obtained through the application of the CS (upper row) and FS (lower row) methods. The involved synthetic datasets: SD3 (SUM inference method, first feature), SD5 (MAX inference method, first feature), SD6 (SUM inference method, first feature).

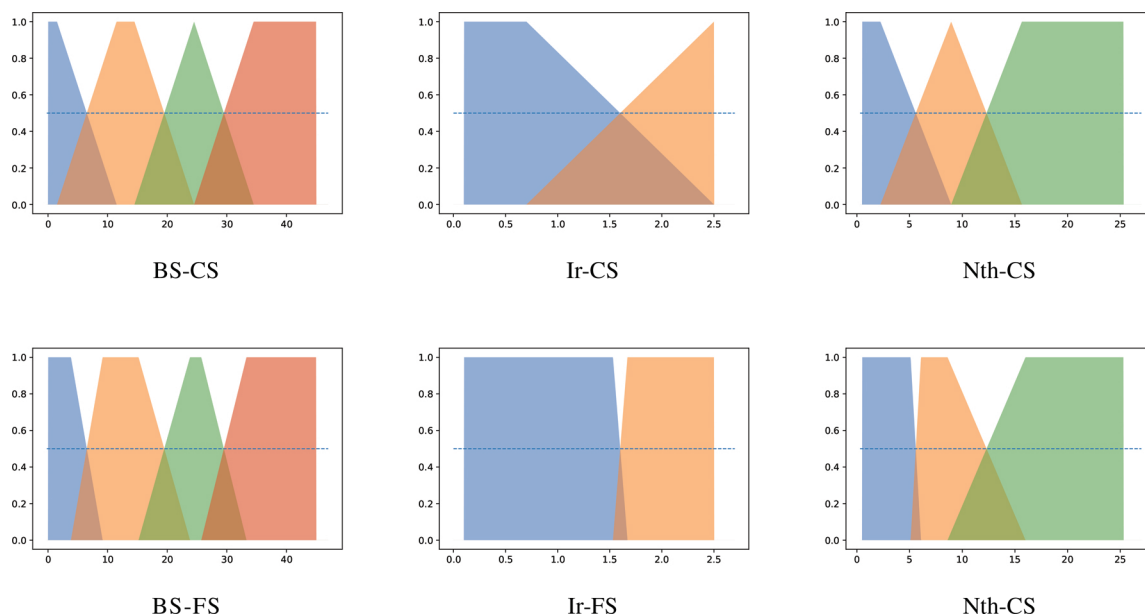


Figure 6 Comparison of the trapezoidal SFPs obtained through the application of the CS (upper row) and FS (lower row) methods. The involved benchmark datasets: BS (SUM inference method, first feature), Ir (MAX inference method, fourth input), Nth (SUM inference method, first feature).

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHORS CONTRIBUTED

All authors read and approved the final manuscript.

ACKNOWLEDGMENTS

The research is partially supported by Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) under grant PON ARS01_00141 "CLOSE." The authors are members of the INdAM Research group GNCS.

REFERENCES

- [1] S. Guillaume, B. Charnomordic, Learning interpretable fuzzy inference systems with fispro, *Inf. Sci.* 181 (2011), 4409–4427.
- [2] J.M. Alonso, C. Castiello, M. Lucarelli, C. Mencar, Modeling interpretable fuzzy rule-based classifiers for Medical Decision Support, in: R. Magdalena, E. Soria, J. Guerrero, J. Gómez-Sanchis, and A. Serrano (Eds.), *Medical applications of intelligent data analysis: research advancements*, IGI Global, Hershey, PA, 2012, pp. 255–272.
- [3] J.M. Alonso, C. Castiello, C. Mencar, The role of interpretable fuzzy systems in designing cognitive cities, in: E. Portmann, R. Seising, M. Tabachi (Eds.), *Designing cognitive cities: linking citizens to computational intelligence to make efficient, sustainable*

- and resilient cities a reality, vol. 176, Springer Verlag, 2019, pp. 131–152.
- [4] A. Fernandez, F. Herrera, O. Cordon, M. Jose del Jesus, F. Marcelloni, Evolutionary fuzzy systems for explainable artificial intelligence: why, when, what for, and where to? *IEEE Comput. Intell. Mag.* 14 (2019), 69–81.
 - [5] J. Casillas, O. Cordón, F.H. Triguero, L. Magdalena, Interpretability issues in fuzzy modeling, vol. 128, Springer, Berlin Heidelberg, 2013.
 - [6] F. Rudziński, A multi-objective genetic optimization of interpretability-oriented fuzzy rule-based classifiers, *Appl. Soft Comput.* 38 (2016), 118–133.
 - [7] D.P. Rini, S.M. Shamsuddin, S.S. Yuhaniz, Particle swarm optimization for anfis interpretability and accuracy, *Soft Comput.* 20 (2016), 251–262.
 - [8] D. Wang, C. Quek, A.H. Tan, C. Miao, G.S. Ng, Y. Zhou, Leveraging the trade-off between accuracy and interpretability in a hybrid intelligent system, in: 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), IEEE, 2017, pp. 55–60.
 - [9] A.E. Yankovskaya, I. Gorbunov, I. Hodashinsky, Tradeoff search methods between interpretability and accuracy of the identification fuzzy systems based on rules, *Pattern Recognition Image Anal.* 27 (2017), 243–265.
 - [10] M.I. Rey, M. Galende, M.J. Fuente, G.I. Sainz-Palmero Multi-objective based fuzzy rule based systems (FRBSs) for trade-off improvement in accuracy and interpretability: A rule relevance point of view, *Knowledge Based Syst.* 127 (2017), 67–84.
 - [11] L. Magdalena, Semantic interpretability in hierarchical fuzzy systems: creating semantically decouplable hierarchies, *Inf. Sci.* 496 (2019), 109–123.
 - [12] J.M. Alonso, C. Castiello, C. Mencar, Interpretability of fuzzy systems: current research trends and prospects, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015, pp. 219–237.
 - [13] F. Masulli, S. Rovetta, The challenges of big data and the contribution of fuzzy logic, in: R. Fullér, S. Giove, F. Masulli (Eds.), *Fuzzy logic and applications*, Springer International Publishing, Cham, 2019, pp. 261–264.
 - [14] K. Cpałka, *Design of Interpretable Fuzzy Systems*, vol. 684, Springer, Switzerland, 2017.
 - [15] J.M. Alonso, From Zadeh's computing with words towards explainable artificial intelligence, in: *International Workshop on Fuzzy Logic and Applications*, Springer, Cham, 2018, pp. 244–248.
 - [16] K. Nozaki, H. Ishibuchi, H. Tanaka, Adaptive fuzzy rule-based classification systems, *IEEE Trans. Fuzzy Syst.* 4 (1996), 238–250.
 - [17] J.A. Roubos, M. Setnes, J. Abonyi, Learning fuzzy classification rules from labeled data, *Inf. Sci.* 150 (2003), 77–93.
 - [18] P.P. Angelov, X. Zhou, Evolving fuzzy-rule-based classifiers from data streams, *IEEE Trans. Fuzzy Syst.* 16 (2008), 1462–1475.
 - [19] C. Mencar, C. Castiello, R. Cannone, A.M. Fanelli, Design of fuzzy rule-based classifiers with semantic cointension, *Inf. Sci.* 181 (2011), 4361–4377.
 - [20] C.S. Herrmann, Symbolical reasoning about numerical data: a hybrid approach, *Appl. Intell.* 7 (1997), 339–354.
 - [21] D.E.A. Giles, R. Draeseke, Econometric Modelling based on Pattern recognition via the Fuzzy c-Means Clustering Algorithm, *Econometrics Working Papers 0101*, Department of Economics, University of Victoria, 2001.
 - [22] H. Wang, J. Xiao, TS fuzzy system based on multi-resolution analysis and its function approximation, in: *Fifth World Congress on Intelligent Control and Automation*, vol. 1, IEEE, 2004, pp. 244–249.
 - [23] K. Loquin, O. Strauss, Fuzzy histograms and density estimation, in: *Soft methods for integrated uncertainty modelling*, Springer, Berlin, Heidelberg, 2006, pp. 45–52.
 - [24] S. Weber, Measures of fuzzy sets and measures of fuzziness, *Fuzzy Sets Syst.* 13 (1984), 247–271.
 - [25] J. Dombi, L. Porkoláb, Measures of fuzziness, *Ann. Univ. Scientiarum Budapestinensis Sectio Computatorica.* 12 (1991), 69–78.
 - [26] P.Y. Glorenc, Constrained optimization of fuzzy decision trees, in: *Interpretability issues in fuzzy modeling*, Springer, 2003, pp. 125–147.
 - [27] A. Segatori, F. Marcelloni, W. Pedrycz, On distributed fuzzy decision trees for big data, *IEEE Trans. Fuzzy Syst.* 26 (2017), 174–192.
 - [28] C.Y. Teh, Y.W. Kerk, K.M. Tay, C.P. Lim, On modeling of data-driven monotone zero-order tsf fuzzy inference systems using a system identification framework, *IEEE Trans. Fuzzy Syst.* 26 (2018), 3860–3874.
 - [29] M. Barsacchi, A. Bechini, P. Ducange, F. Marcelloni, Optimizing partition granularity, membership function parameters, and rule bases of fuzzy classifiers for big data by a multi-objective evolutionary approach, *Cognitive Comput.*, 11, pp. 1–21.
 - [30] C. Mencar, M. Lucarelli, C. Castiello, A.M. Fanelli, Design of strong fuzzy partitions from cuts, in: *Proc. 8th Conf. of the European Society for Fuzzy Logic and Technology (EUSFLAT 2013)*, Milan, Italy, 2013, pp. 424–431.
 - [31] C. Castiello, C. Mencar, Exploiting particle swarm optimization to attune strong fuzzy partitions based on cuts, in: *2019 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (EUSFLAT 2019)*, Atlantis Press, 2019.
 - [32] Y.C. Hu, R.S. Chen, G.H. Tzeng, Discovering fuzzy association rules using fuzzy partition methods, *Knowledge Based Syst.* 16 (2003), 137–147.
 - [33] O. Cordón, A historical review of evolutionary learning methods for mamdani-type fuzzy rule-based systems: designing interpretable genetic fuzzy systems, *Int. J. Approximate Reasoning.* 52 (2011), 894–913.
 - [34] C.T. Sun, Rule-base structure identification in an adaptive-network-based fuzzy inference system, *IEEE Trans. Fuzzy Syst.* 2 (1994), 64–73.
 - [35] I. Cloete, J.M. Zurada, *Knowledge-based neurocomputing*, The MIT Press, Cambridge, MA, 2000.
 - [36] O. Cordón, F. Herrera, F. Hoffmann, L. Magdalena, *Genetic Fuzzy Systems*, World Scientific, 2001.
 - [37] G. Castellano, C. Castiello, A.M. Fanelli, L. Jain, Evolutionary neuro-fuzzy systems and applications, in: *Advances in evolutionary computing for system design*, Springer, 2007, pp. 11–45.
 - [38] J.M. Alonso, C. Castiello, C. Mencar, Interpretability of fuzzy systems: current research trends and prospects, in: *Springer handbook of computational intelligence*, Springer, Berlin, Heidelberg, 2015, pp. 219–237.
 - [39] R. Alcalá, J. Alcalá-Fdez, M.J. Gacto, F. Herrera, Rule base reduction and genetic tuning of fuzzy systems based on the linguistic 3-tuples representation, *Soft Comput.* 11 (2007), 401–419.
 - [40] J.A. Sanz, A. Fernández, H. Bustince, F. Herrera, Improving the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets and genetic amplitude tuning, *Inf. Sci.* 180 (2010), 3674–3685.

- [41] C.H.Nguyen, T.S. Tran, D.P. Pham, Modeling of a semantics core of linguistic terms based on an extension of hedge algebra semantics and its application, *Knowledge Based Syst.* 67 (2014), 244–262.
- [42] R. Munoz-Salinas, E. Aguirre, O. Cordón, M. Garcia-Silvente, Automatic tuning of a fuzzy visual system using evolutionary algorithms: single-objective versus multiobjective approaches, *IEEE Trans. Fuzzy Syst.* 16 (2008), 485–501.
- [43] R. Poli, J. Kennedy, T. Blackwell, Particle swarm optimization. an overview, *Swarm Intell.* 1 (2007) 33–57.
- [44] J. Kenn, R. Eberhart, Particle swarm optimization, in: *Proc. International Conference on Neural Networks (ICNN'95)*, vol. 4, Perth, WA, Australia, 1995, pp. 1942–1948.
- [45] Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, in: V. Porto, N. Saravanan, D. Waagen, A. Eiben (Eds.), *Evolutionary Programming VII. EP 1998*, vol. 1447, Springer, Berlin, Heidelberg, 1998, pp. 591–600.
- [46] J. Kennedy, R.C. Eberhart, Y. Shi, *Swarm intelligence*, Morgan Kaufmann, San Francisco, CA, 2001.
- [47] M.Clerc, J. Kennedy, The particle swarm—explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans. Evol. Comput.* 6 (2002), 58–73.
- [48] R. Mendes, J. Kennedy, J. Neves, Watch thy neighbor or how the swarm can learn from its environment, in: V. Porto, N. Saravanan, D. Waagen, A. Eiben (Eds.), *Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS'03)*, IEEE, 2003, pp. 88–94.
- [49] R.C. Eberhart, Y. Shi, Comparison between genetic algorithms and particle swarm optimization, in: V. Porto, N. Saravanan, D. Waagen, and A. Eiben (Eds.), *Evolutionary Programming VII. EP 1998*, vol. 1447, Springer, Berlin, Heidelberg, 1998, pp. 611–616.
- [50] J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, *J. Multiple Valued Logic Soft Comput.* 17 (2011), 255–287.
- [51] D. Dua, C. Graff, UCI machine learning repository, <http://archive.ics.uci.edu/ml>, 2017.
- [52] C. Castiello, Beer dataset beerstyles3features-1.0 (2017).
- [53] C. Castiello, A.M. Fanelli, M. Lucarelli, C. Mencar, Interpretable fuzzy partitioning of classified data with variable granularity, *AAppl. Soft Comput.* 74 (2019), 567–582.
- [54] R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1, 2000, pp. 84–88.
- [55] I.C. Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, *Inf. Process. Lett.* 85 (2003), 317–325.
- [56] M. Xu, J. Gu, Parameter selection for particle swarm optimization based on stochastic multi-objective optimization, in: *2015 Chinese Automation Congress (CAC)*, 2015, pp. 2074–2079.