University of Alberta

System Modeling with Granular Architectures of Computational Intelligence

by

Mingli Song

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering

Fall 2012
Edmonton, Alberta

Abstract

Information granules regarded as the key components of knowledge representation help generalize information while the level of granularity of information granules becomes crucial to the problem description and an overall strategy of problem solving in system modeling. The ultimate challenge is to develop a comprehensive model within which information granules serving as an important design asset helps realize problem solving.

In this study, we propose a system modeling framework of granular architectures and evaluate its effectiveness. We develop a new approach to build functional rule-based fuzzy models by focusing on the reduction of input space which is realized by genetic algorithms. A concept and practice of a granular fuzzy model is established with interval results of global character by aggregating some collective sources of knowledge (local models). The granular fuzzy model based on interval analysis directly reflects upon the diversity of the local sources of knowledge in which intervals are constructed through the use of the principle of justifiable granularity.

Two approaches are proposed to design granular neural networks. The first method is concerned with a formation of a global granular neural network whose architecture is formed as a result of reconciliation of outcomes produced by local neural networks. The second method is aimed at the realization of granular neural networks through the formation of interval (granular) connections around numeric connections of the original neural networks where single- and multiple-objective particle swarm optimization is used.

We develop a granular analytic hierarchy process (AHP), which provides decision--makers a significant level of flexibility (expressed by the granular nature of the underling construct) so that their initial preferences can be adjusted within a certain interval to

achieve higher level of consensus within the group. Moreover, a granulation of linguistic information used in the AHP model is adopted to elevate the consistency of the obtained solution.

Throughout the overall study, particle swarm optimization is used as a comprehensive optimization framework to realize the design of granular constructs.

Acknowledgements

Foremost, I would like to express my deepest gratitude and thanks to my supervisor, Dr. Witold Pedrycz, for his indispensable and invaluable guidance. He provides me continuous support and encouragement during the whole research process, brings me a spirit of exploring science to the real world application. The completion of this thesis would be impossible without his close supervision and devotion. His patience, innovations, enthusiasm, and prestigious contributions to Software Engineering and Intelligent Systems make him a great mentor.

Additionally, I want to give my sincere thanks to my parents, Yun Pan and Xidi Song; my boyfriend, Dunming Xu; my sister Mingming Song, and all my friends. Thanks for their support, trust and encouragement all along, which help me to complete four-year's Ph.D. study smoothly.

Table of Contents

List of Tables

List of Figures

Common nomenclature

symbol	description
N	number of data
$\mathbf{x} = [x_1, x_2, \ldots, x_n]$	an instance in a data set with n features
target	output of an instance
c	number of clusters
$U = [u_{ik}]$, $i = 1, 2, .., c$, $k = 1,2,\ldots,N$	partition matrix
$V = [\mathbf{v}_i]$	prototypes
m	fuzzification coefficient
σ	standard deviation
Q	objective function value
ν	inconsistency index
λ	the largest eigenvalue of a matrix
$\mathbf{e}$	eigenvector corresponding to the largest eigenvalue
$\mathbf{s} = [s_{1,} s_2, \ldots, s_2]$	position vector in PSO
$\mathbf{v} = [v_{1,} v_2, \ldots, v_2]$	velocity vector in PSO
η	inertia weight
$\boldsymbol{G}(\mathbf{R})$	a granular construct
I	coverage ratio
L	average length of intervals
w	weight of a neural network

1. Introduction

1.1 Motivation

Computational Intelligence (CI) is a field of intelligent information processing related with different branches of computer science and engineering (Pedrycz and Gomide, Fuzzy Systems Engineering: toward human-centric computing 2007). The core of CI embraces fuzzy systems, neural networks, and evolutionary computation and its ultimate aim is dealing with various ways in which these technologies are brought together to constitute some form of hybrid architectures. System modeling plays a very important role in many areas such as control, expert systems, communications, etc. One can convincingly note that reaching the ultimate agenda of system modeling—approximation, optimization, prediction, decision making and alike requires not only intelligent information processing techniques but also some flexibility and elasticity within the models to adjust their architectures and the corresponding learning algorithm.

The purpose of system modeling is to model the operation of an unknown system from a set of measured input–output data. While many approaches have been proposed to deal with this problem, several outstanding questions still remain that need to be addressed in this field. The first one is that many existing algorithms achieve high accuracy of their models at the expense of the complexity. It is hard to be avoided since generally more information helps reveal more details about the results. Thus how to balance these two factors becomes crucial. The second challenge is concerned with exploration of mechanisms for developing a model. It becomes very difficult to develop a system model when the unknown system is highly nonlinear and complex. Third, the selection of structure and parameters identification methods is a considerable challenge. Though in many researches the authors give effective solutions for a particular problem, there is no general design framework to solve this problem.

Information granules regarded as conceptually justifiable constructs constitute building blocks of intelligent systems. As such, information granules are used in various pursuits of system analysis and design. However, the studies reported so far concentrate on proposing granular architectures at the very beginning which makes the subsequent learning process more or less sophisticated.

The above observations and findings motivate the following studies on system modeling issues. Intuitively, if we allow any model to treat their entries (parameters or input or output) not as numeric values but rather as information granules, this will bring a badly required factor of flexibility. In this manner, the model itself may admit all numeric realizations, which are compatible with the more general granular abstraction. We introduce a concept of a granular model with this context and emphasize a role of information granularity being regarded here as an important conceptual and computational resources which can be exploited as a means to improve the performance of the model in terms of some evaluation criterion. In a nutshell, the level of granularity could be treated as synonymous of the level of flexibility injected into the modeling environment. This terminology underlines an important, unique role being played by information granules. By admitting a certain level of information granularity, we are provided with a unique possibility to navigate in the space of entries assigned to a model.

1.2 Objectives and Main Contributions

The main concepts of this study can be outlined and illustrated in the following graph.

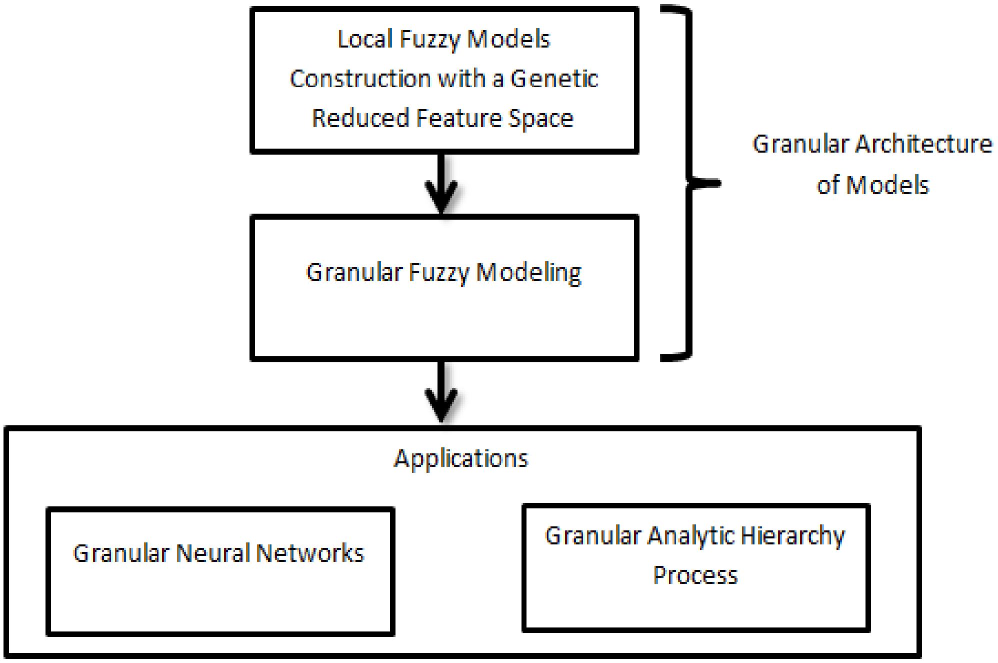

Local Fuzzy Models Construction with a Genetic Reduced Feature Space
Granular Fuzzy Modeling
Granular Architecture of Models
Applications
Granular Neural Networks
Granular Analytic Hierarchy Process

Figure 1. The main concepts of the dissertation

The aim of this study is to address these challenges present in system modeling by developing models with granular architectures capable of capturing essence of domain knowledge to benefit the modeling. Specifically, these objectives are to

- construct local functional rule-based fuzzy models by focusing on the reduction of input space which is realized by genetic algorithms

- establish a concept and practice of granular fuzzy models with interval results of global character by aggregating some collective sources of knowledge (local models)
- form a global granular neural network whose architecture is formed as a result of reconciliation of outcomes produced by local neural networks
- build granular neural networks through the formation of interval (granular) connections around numeric connections of the original neural networks which is realized by single- and multiple-objective particle swarm optimization
- develop a granular analytic hierarchy process to achieve higher level of consensus within the group
- adopt a granulation of linguistic information used in the AHP model to elevate the consistency of the obtained solution

It is interesting and practically legitimate to consider realizing system modeling with granular architecture of CI. CI being commonly viewed as a highly synergistic framework of neurocomputing, fuzzy sets, and evolutionary optimization benefits immensely from the concept of information granularity, cf. (Karray and Silva 2004) (Pedrycz, Computational Intelligence: An Introduction 1998). The idea of granular constructs of CI brings forward an essential generalization aspect of fundamental architectures, such as neural networks, fuzzy models (classifiers, predictors, etc.) that result in granular neural networks and granular fuzzy models. Likewise, we name the generalized model no matter with what architectures a granular model due to the granular information used to build the model.

Information granulation also becomes beneficial to the process of knowledge management in a multimodel CI environment (say, multi-agent systems) by facilitating the realization of knowledge sharing, reconciliation, consensus building, and knowledge transfer, cf. (Pedrycz and Rai, A multifacted perspective at data analysis: a study in collaborative intelligent agents 2008). In this sense, information granularity emerges as an immediate and inevitable consequence of the existing diversity of sources of knowledge (models) involved in the overall process of knowledge management. The granularity of the resulting construct is essential to the description and quantification of diversity and coherence of the available sources of knowledge. Information granularity furnishes a very much-needed level of flexibility that becomes crucial to facilitate interaction.

The formation of information granules can be realized by using the principle of justifiable granularity. In social computing, we are faced with a number of separate streams of information granules generated by individual sources (users, models) and the resulting individual sources of knowledge form on their basis. An ultimate objective is to realize an effective interaction at the global basis by invoking some mechanisms of knowledge sharing and collaboration. In this way, each source of knowledge is developed not only by relying on some experimental evidence that becomes locally available but is exposed to some general conceptual perspective by effectively communicating with other entities and sharing and reconciling.

Since there are various diversities of selection of frameworks of the individual sources, the building strategies of a granular model for the entire system or phenomenon may differ a lot. We devote ourselves to architectures of neural networks, fuzzy models and decision making models on basis of information granulation in which knowledge sharing, reconciliation and knowledge transfer are incorporated among individual sources of knowledge. The formation of local models, knowledge transfer and collaboration among them, and their effect on the final outcomes of the overall model are elaborated in all cases.

Being treated as an important design asset, the assumed level of information granularity being distributed (allocated) among an architecture in a certain way realizes a formation of an abstract and general architecture. Neural networks as a typical example cf. (Ishibuchi and Tanaka, An architecture of neural networks with interval weights and its application to fuzzy regression analysis 1993) (Ishibuchi, Kwon and Tanaka, A learning algorithm of fuzzy neural networks with triangular fuzzy weights 1995) (Park, Pedrycz and Oh 2009) (Zhang, Jin and Tang, Granular neural networks with evolutionary interval learning 2008) (Zhang, Fraser, et al. 2000) are generalized through allocation of granularity to all connections (weights and biases). The obtained granular structure reflects the general nature of a system.

A phenomenon that cannot be overlooked in system modeling is the pre-processing of input data, especially when dealing with data with high dimensionality. Some effective methods have been proposed. Hence we choose one of them called feature reduction which aims at reducing the searching space of input data as well as making the model more transparent (interpretable) especially when doing fuzzy modeling. The study tells a truth that an effective pre-processing technique may improve a lot the performance of a model in terms of structure and the learning algorithm.

System modeling in solving a group decision making problem, (DeSanctis and Gallupe 1987), is always challenging. This is due to the nature of the problem: pursuit of reconciling differences of opinions (judgments) expressed by individual members of the group; once an individual changes his/her idea, others have to change simultaneously. Fuzzy decision making mechanisms bring a great deal of flexibility, cf. (Prodanovic and Simonovic 2003) (Herrera-Viedma, Alonso, et al. 2007) (Mata, Martinez and Herrera-Viedma 2009). By admitting membership degrees, we are offered flexibility to exploit different aggregation mechanisms and navigate a process of interaction among decision-makers to achieve increasing level of consistency within the group. While the studies reported so far exploit more or less sophisticated ways of adjusting initial judgments (preferences) of individuals, here we bring forward a concept of information granularity. It is viewed as an essential asset, which offers a decision-maker a tangible level of flexibility using which some initial preferences conveyed by each individual can be adjusted with intent of reaching a higher level of consensus. This again testifies the profound role of information granules in system modeling.

To be fully utilized, linguistic information present in decision making problems (Bordogna, Fedrizzi and Pasi 1997) has to be made operational through information granulation. Moreover, in general, information provided by humans is inherently non-numeric in decision-making problems. This phenomenon makes us look for some ways to map linguistic terms to numeric representations. Information granules and information granulation occur here as an overwhelming alternative. The mapping of the linguistic terms to the predetermined scale produces a linguistic decision matrix and the consistency of it can be regarded as an optimization object. Finally we reach the same goal: increase consistency of an individual decision maker or consensus among decision makers.

The main contributions of this study are

- Various classic and well-studied models and the corresponding learning algorithms form the basic framework of this research. Fuzzy models, neural networks and analytic hierarchy process which are used for the aim of approximation, optimization, prediction, decision making are for instance.
- The usage of information granules as an important modeling asset is the second accomplishment.
- A novel way of feature reduction in data pre-processing realized by genetic algorithm can be seemed as a third contribution.
- The fourth novelty is in the knowledge management and reconciliation among individuals models used to build a granular model.
- A model with granular architecture that is trained through using some optimization technique and reaches optimal in terms of a certain criterion is formulated.

Some benchmark data sets from Machine Learning website (http://archive.ics.uci.edu/ml/datasets.html) and StatLib website (http://lib.stat.cmu.edu/datasets/) are used to provide a detailed insight into the algorithm or developed model. Those data sets are: Auto MPG, Boston Housing, Concrete compressive strength, Abalone, PM10, Bodyfat and so on. Their detailed introduction is elaborated when the first time it is used.

The notation and nomenclature used here are the ones being commonly encountered in literature.

1.3 Dissertation Organization

The subsequent chapters are structured as follows.

Chapter 2 Background and literature review
Some fundamentals about system modeling along with their development in literature are introduced: the conceptual and algorithmic underpinnings of information granules, information granulation, computing with information granules; classic system modeling methods, analytic hierarchy process in particular; particle swarm optimization and genetic algorithms as critical assistant tools of population based optimization methods.

Chapter 3 The principle of justifiable granularity
This chapter is concerned with the principle of justifiable granularity which is used to form a meaningful representation of a collection of numeric values. We show a construction of interval-based information granules as well as information granules represented as fuzzy sets. The principle of information granularity becomes one of key techniques of building granular models.

Chapter 4 Allocation of information granularity
Allocation of information granularity is another key technique we use to develop granular models. Four common protocols are discussed along with illustrative examples: in forming granular neural networks and in group decision making problems.

Chapter 5 A genetic reduction of feature space in the design of local fuzzy models
In this chapter, an effective genetic reduction of input space method is discussed and compared with the results generated by no feature reduction in fuzzy modeling. Evolutionary methods are employed to reduce features. Synthetic data, auto MPG data set and Bodyfat data set are used to test our approach.

Chapter 6 Granular fuzzy modeling
We propose an advanced fuzzy model (a granular fuzzy model): a hierarchy fuzzy model aggregated by local fuzzy models in which the principle of information granularity applies in this chapter. The local fuzzy models are realized by a commonly used type: linear functional fuzzy models. Numeric results (by using concrete compressive strength data set and PM10 data set) are discussed with intent of displaying the most essential features of the proposed methodology and algorithmic developments.

Chapter 7 From local neural networks to granular neural networks

Here we are concerned with building granular models on another fundamental architecture of system modeling—neural networks (a set of local neural networks) which also appears as a core of CI. The essential design technique is the principle of justifiable granularity. Three data sets are used here: abalone data set, Auto MPG data set and Boston Housing data set.

Chapter 8 Local granular neural networks

In this chapter, we propose an idea of building a granular neural network based on a single network. Thus the development process embraces mainly two steps: train a neural network; allocate information granularity to its connections and form granular connections. A series of numeric studies completed for synthetic data, auto MPG, Boston Housing, PM10 and Bodyfat data sets provide a useful insight into the effectiveness of the proposed algorithm.

Chapter 9 Granularity of information in solving group decision-making problems

Chapter 9 is about solving the group decision making problems with analytic hierarchy process and allocation of information granularity. A number of numeric studies are provided to illustrate an essence of our method.

Chapter 10 Granulation of linguistic information in decision making

A commonly faced phenomenon present in decision-making--linguistic information has to be made operational through information granulation in this chapter. Both individual and group decision-making models of analytic hierarchy process are discussed. Particle swarm optimization is used as an optimization framework.

Chapter 11 Conclusions

In this chapter, we draw conclusions from our works. We also consider the future work in the area of system modeling with granular architectures.

2. Background and Literature Review

In this chapter, background knowledge of information granules and literature review of traditional system modeling approaches are given. Firstly a brief introduction to granular information is provided in which we emphasize several formalisms, their characterizations, design process and operations.

Following the introduction of information granulation, the basic frameworks, architectures and algorithms in system modeling are outlined to offer a basis on which to discuss a model with information granules in subsequent chapters. Two well-known modeling architectures which are usually used to model complex systems are reviewed: fuzzy models and neural networks.

We recall one of the most popular strategies in solving decision making problems—analytic hierarchy process (AHP). Some recent developments of AHP are also given.

Due to the complexity of optimization problems encountered in various modeling algorithms, a selection of a proper optimization vehicle becomes crucial. Developments in particle swarm algorithm since its origin along with its benefits and drawbacks are mainly discussed as particle swarm optimization provides a simple realization mechanism and high convergence speed.

2.1. Granular Information

Information granules, as the name itself stipulates, are collections of entities, usually originating at the numeric level, that are arranged together due to their similarity, functional adjacency, and coherency or alike. Information granules are seen everywhere: spatial granulation like image processing and Geographic Information System (GIS), temporal granulation like cultural, legal, business orientation of the designer and so on. They are central to processes of abstraction which guide our intellectual pursuits. The process of constructing information granules is referred to as information granulation. The notion of granulation was introduced to deal with information abstraction and summarization typically in data related works, cf. (Zadeh, Fuzzy sets and information granularity, in: M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.) 1979) (Pedrycz, Granular Computing: An emerging paradigm 2001). It is very important and necessary since too detailed information is rarely processed due to some physical and cognitive limitations, and user's requirements.

Quite often, we need a reduced number of variables and instances who own representative characters of the whole data or satisfy some requirements by the particular task. To solve this problem, a procurement of effective abstraction is needed. Information granules can be used to represent the aggregated detailed numeric information. The formed information granules are regarded as a collection of elements that are perceived as being indistinguishable, similar close or functionally equivalent. We below first take a quick browse of several representation formalisms of information granules.

Granular computing is knowledge-oriented and the knowledge-inclined processing arises as a cornerstone of data mining, rule-based models, hierarchical and supervisory control or decision making systems, etc. Hence, some fundamental algorithms within granular computing are introduced which will be often used in system modeling.

2.1.1. Representations of information granules

There are some different formalisms and concepts of information granules, for example, sets (intervals), fuzzy sets, rough sets, etc. cf. (Bargiela and Pedrycz 2003) (Pedrycz and Gomide, Fuzzy Systems Engineering: toward human-centric computing 2007). We will first review some basic concepts and operations on these information granules.

A. Sets (Intervals)

Set theory occupies an important and unique place in modern mathematics since it can be shown that it can be used as a starting point for the derivation of all other branches of mathematics. In set theory, interval arithmetic which first appeared in (Warmus 1956) (Warmus 1961) and (Sunaga 1958) offered an important generalization of arithmetic defined on real numbers. Since then there has been a sustained research effort in developing theory, applications and the support software for interval computations, cf. (Kearfott and Kreinovich 1996) (Kreinovich, et al. 1998). One example of the most successful application of intervals to solve real-life problems is that of solving Gibbs free-energy equations arising in a super-conducting super-collider design (Stadtherr, Schnepper and Brennecke 1994). The significance of that solution is twofold: one is that it showed the computational feasibility of large scale interval computations and the other is it provided a benchmark against which many floating-point solutions were measured.

Owing to the ease of manipulation of intervals and their straightforward interpretation, we mostly use intervals as a formal format of information granules in our research. Each numeric interval includes a collection of elements in the line of real numbers and is bounded by two numeric values: lower bound and upper bound. Multidimensional constructs are built upon Cartesian products of numeric intervals. The operations on intervals are also easy to realize. This makes itself preferred by many people.

Now let's recall some results of interval operations. Given information granules represented in the form of intervals, say X = [a, b], Y = [c, d], etc., the algebraic operations are expressed as follows (de Weerdt, Chu and Mulder 2009):

Addition: X + Y = [a + c, b + d]; (2.1)
Subtraction: X - Y = [a - d, b - c]; (2.2)
Multiplication: $X \times Y$ = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)] (2.3)
Division: (excluding division by an interval containing 0)
$\frac{X}{Y} = X \times (\frac{1}{Y})$ with $\frac{1}{Y} = [\frac{1}{d}, \frac{1}{c}]$ (2.4)
Furthermore when it comes to the mapping of intervals, we have the following results.
non-decreasing function f:
f(X) = f([a, b]) = [f(a), f(b)] (2.5)
non-increasing function f:
f(X) = f([a, b]) = [f(b), f(a)] (2.6)

B. Fuzzy Sets

Fuzzy sets provide a possibility to formally express concepts of continuous boundaries. These concepts are everywhere. When expressing ideas, describing concepts and communicating with people, we always use terms to which the yes-no does barely apply. Natural language like low, high, small and alike terms is common environment with huge similar examples of such description. Since early 1900s, researchers began to observe the limitations of dichotomy, for instance, (Duhem 1906), (Wiener 1923), (Korzybski 1933), (Black 1937), etc. In 1965, L. A. Zadeh introduced a concept of "fuzzy set" which partial membership of elements to a set is admitted. Fuzzy set A is generally described by a membership function which maps the universe of discourse X in which A is defined into a unit interval:

$A: X \rightarrow [0, 1]$

Formally, A(x) denotes a degree of membership that describes an extent to which x belongs to A. If A(x) = 1, then we conclude that x is fully included in A. If A(x) = 0, then we know that x is fully excluded from A. The values between 0 and 1 point at a partial membership of X to A. The higher the membership degree is, the stronger the association of the given element to the concept.

For instance, when we talk about temperature, a value of $19\,^{\circ}C$ would be fully compatible with the concept of *comfortable*, yet $0\,^{\circ}C$ and $30\,^{\circ}C$ would not. In this case, a single number does not make too much sense.

(Dubois and Prade 1997) discussed three points of view at fuzzy sets which are closely linked with general areas of applications. In these view, membership grades take specific meanings: degree of similarity, degree of preference and degree of uncertainty.

Membership functions evaluate the notion of partial membership. Formally speaking, any function A: X→ [0, 1] could be qualified to serve as a membership function describing the corresponding fuzzy set. In practice, the form of the membership functions should be reflective of the problem at hand for which we construct fuzzy sets. We can anticipate a diversity of the types of membership functions to represent different problems. For example, triangular functions, trapezoidal functions, Gaussian functions, parabolic functions, non-symmetric Gaussian functions and others, please refer to Figure 2.

Fuzzy sets can be characterized by a scalar index: size (granularity). There are a number of descriptors commonly encountered in practice. We here list one which is going to be used in our following studies—cardinality. The formula is (2.7).

Cardinality: Computing the cardinality is about enumerating (counting) the number of elements in the information granule.

$$\text{Card}\ (A) = \int_X A(\mathbf{x})d\mathbf{x} \tag{2.7}$$

where A is an information granule under consideration. The higher the cardinality is, the higher the abstraction of the granule and the lower its specificity.

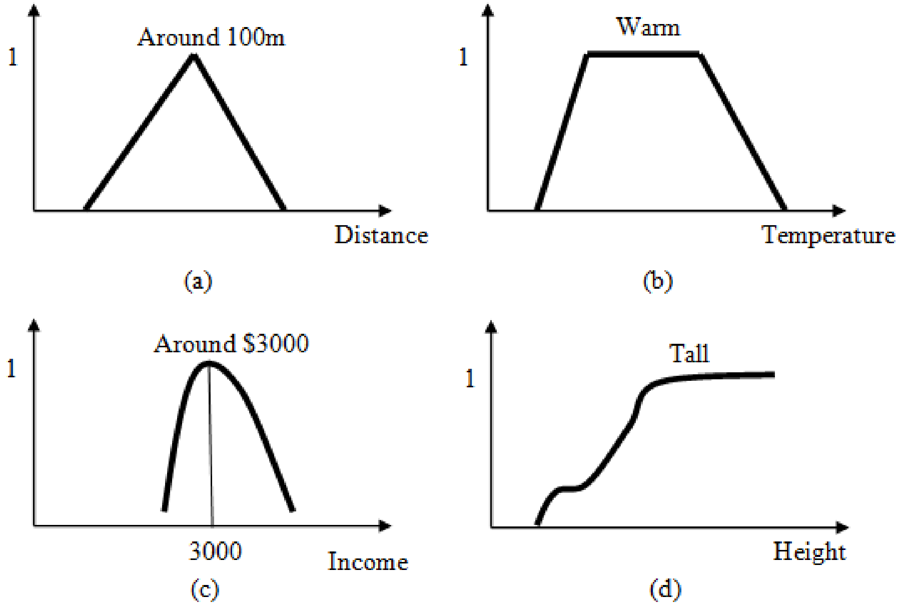


Figure 2. Four types of membership functions: triangular, trapezoidal, Gaussian, Γ-Membership functions

C. Rough Sets

The description of information granules completed with the aid of some vocabulary is usually imprecise. Intuitively, such description may lead to some approximations called lower and upper bounds. This is the essence of rough sets introduced by (Pawlak, Rough sets 1982) (Pawlak 1991); refer also to (Skowron 1989). Interesting generalizations, conceptual insights, and algorithmic investigations are offered in a series of fundamental papers by (Pawlak and Skowron, Rudiments of rough sets 2007) (Pawlak and Skowron, Rough sets and Boolean reasoning 2007) (Pawlak and Skowron, Rough sets: Some extensions 2007). To explain the concept of rough sets and show what they are to offer in terms of representing information granules, we use an illustrative example displayed in Figure 3. Consider a description of environmental conditions expressed in terms of temperature and pressure. For either of these factors, we fix several ranges of possible values where each of such ranges comes with some interpretation such as “values below”, “values in-between”, “values above”, and so on. By admitting such selected ranges in both variables, we construct a grid of concepts formed in the Cartesian product of the spaces of temperature and pressure. In more descriptive terms, this grid forms a vocabulary of generic terms using which we would like to describe all new information granules.

Now let us consider that the environmental conditions monitored over some time have resulted in some values of temperature and pressure ranging in-between some lower and upper bound. Denote this result by X. When describing it in terms of the elements of the vocabulary, we end up with a collection of elements that are fully included in X. They form a lower bound of description of X ($\underline{Apr_A}(X)$) when being completed in presence of the given vocabulary. Likewise, we may identify elements of the vocabulary that have a nonempty overlap with X and in this sense constitute an upper bound ($\overline{Apr_A}(X)$) of the description of the given environmental conditions. Along with the vocabulary, the description forms a certain rough set. Please refer to Figure 3.

We are here concerned with a description of a given concept X realized in the language of a certain collection (vocabulary) of rather generic and simple terms A_1, A_2, ..., A_c. The lower and upper boundaries (approximation) are reflective of the resulting imprecision caused by the conceptual incompatibilities between the concept itself and the existing vocabulary.

It is interesting to note that the vocabulary used in the above construct could comprise information granules being expressed in terms of any other formalism, say fuzzy sets. Quite often we can encounter constructs like rough fuzzy sets and fuzzy rough sets in which both fuzzy sets and rough sets are put together.

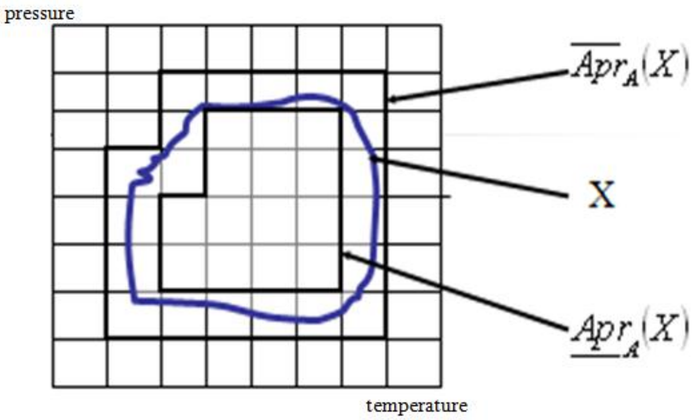


Figure 3. A rough set X and its lower and upper approximation

2.1.2. Fuzzy C-Means (FCM)

Clustering and fuzzy clustering have been regarded as a synonym of structure discovery in data. The result, no matter what technique has been used, comes as a collection of information granules which serve as a quantification of concepts serving as descriptors of the phenomenon behind the data. Fuzzy C-Means is one of the commonly used mechanisms of fuzzy clustering.

Given a collection of n-dimensional data set $\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kn}]$, k = 1, 2, …, N, the task is to determine the structure in the data represented by a collection of c prototypes. Patterns are assigned membership values, i.e. a partition matrix U, to the clusters based on their distance from the prototypes. With FCM, one is concerned with discovering the prototypes and partition matrix U by a minimization of the following objective function Q being regarded as a sum of the squared distances:

$$Q = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{m} \left\| \mathbf{x}_k \text{-} \mathbf{v}_i \right\|^2 \tag{2.8}$$

where $\mathbf{v}_i$'s are n-dimensional prototypes of the clusters, i = 1, 2, ..., c, and $U = [u_{ik}]$ stands for a partition matrix; u_{ik} is the membership degree of data $\mathbf{x}_k$ in the ith cluster. The distance between the data and the prototype $\mathbf{v}_i$ is denoted by $\| \bullet \|$. In this study, the distance function used in the clustering is the standard Euclidean in which the corresponding features are normalized by including the corresponding variance, i.e.

$$\left\| \mathbf{x} - \mathbf{y} \right\|^2 = \sum_{j=1}^{n} (x_j - y_j)^2 / \sigma_j^2 \tag{2.9}$$

where σ_j^2 is the variance of the j-th input variable (feature). The fuzzification coefficient m (>0) expresses the impact of the membership grades on the individual clusters.

The optimization process is completed through a sequence of iterations where we start from some random allocation of data (a certain randomly initialized partition matrix) and carry out updates by adjusting values of the partition matrix and the prototypes through (2.10)(2.11).

$$u_{ik}(t+1) = \frac{1}{\sum_{j=1}^{c}\left(\frac{\left\|\mathbf{x}_k - \mathbf{v}_i(t)\right\|}{\left\|\mathbf{x}_k - \mathbf{v}_j(t)\right\|}\right)^{2/(m-1)}} \tag{2.10}$$

$$\mathbf{v}_i(t) = \frac{\sum_{k=1}^{N} u_{ik}^{m}(t)\mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}(t)} \tag{2.11}$$

The iteration stops when a certain termination criterion has been satisfied. Typically, the termination condition is quantified by looking at the changes in the membership values of the successive partition matrices.

The vast amount of accurate of numerical data available in all domains of human activity highlighted the fact that there is an urgent need for a more human-like processing of information, namely generalization and abstraction. Zadeh first coined the term of information granulation in the framework of fuzzy sets and emphasized the fact that the plethora of details does not amount to knowledge. The ubiquitous nature of information granulation stems from the continuing human endeavour to extract and organize knowledge about the external world for the purpose of decision-making, control, system description, prediction and others.

2.2. Modeling Methods of Complex Systems

The problem of system modeling has been studied in different frameworks, investigated from different standard points, and facilitated by various optimization techniques. System is regarded as a set of interacting or interdependent components forming an integrated whole. A system is a set of elements (often called *'components'* instead) and relationships which are different from relationships of the set or its elements to other elements or sets. Most systems share common characteristics, including: structure, behaviour, interconnectivity. In the process of system modeling, some fundamental architecture, such as artificial neural networks, fuzzy logic, expert system, and statistical methodologies have been vastly developed and applied. Different methods have different advantages and emphasis. Among them, neural networks are proposed to realize high accuracy of modeling in which any non-linear functions can be approximated. Fuzzy models, from the perspective of interpretability, are preferred in many human-centric systems. In what follows, we concentrate on these two examples: fuzzy models and artificial neural networks.

2.2.1. Fuzzy models

Fuzzy models, in one way or another, rely on the use of information granules – fuzzy sets in the formation of their structure (Alcala, et al. 2009) (Delgado, Gomez-Skarmeta and Martin 1997) (Park, Pedrycz and Oh 2009). Information granules constitute a backbone of these models based on which further details are developed. In this way, they constitute highly abstract and flexible constructs. Given the environment of physical variables describing the surrounding world and an abstract view of the system under modeling, a very general picture of the architecture of a fuzzy model (Pedrycz and Gomide, Fuzzy Systems Engineering: toward human-centric computing 2007) is portrayed as present in Figure 4.

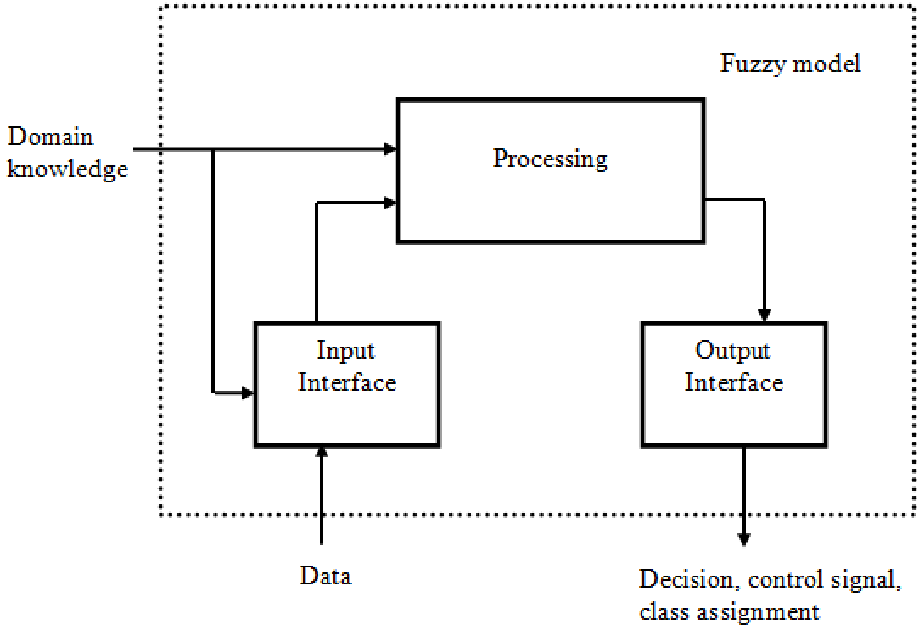


Figure 4. A general view at the underlying architecture of fuzzy models

The three functional components shown in Figure 4 come with different objectives. The input interface builds a collection of modalities (fuzzy sets and fuzzy relations) that are required to link the fuzzy model and its processing core with the external world. This processing core realizes all computing being carried out at the level of fuzzy sets. The output interface converts the results of granular processing into the format acceptable by the external environment.

For instance, in rule-based models a great deal of attention is given to the fuzzy sets forming the condition parts of the consecutive rules. The generic models in this category are formulated as follows:

$$\text{if } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and ... then } y \text{ is } C$$

or

$$\text{if } \mathbf{x} \text{ is } A_i \text{ then } y = f_i(\mathbf{x}, \mathbf{a}_i)$$

where x_1, x_2... are input variables and y is the output variable, whereas A_i, C, ... are the fuzzy sets defined in the corresponding spaces, $f_i(\mathbf{x_i}, \mathbf{a_i})$ denotes a multivariable function, $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ is the vector of base variables and $\mathbf{a}_i$ is a vector of parameters. In particular, one can envision a linear form of the function in which f_i becomes a linear function of its parameters, namely $f_i(\mathbf{x}, \mathbf{a}_i) = \mathbf{a}_i^T\mathbf{x}$. In this case, the inputs are transformed in terms of fuzzy sets and the reasoning result is offered in its numeric format. Rule-based models are central architectures of fuzzy models.

The landscape of fuzzy models is highly diversified. There are several categories of models where each class of the constructs comes with interesting topologies, functional characteristics, learning capabilities and the mechanisms of knowledge representation. Here we list some of them:

- Tabular fuzzy models
- Rule-based models
- Fuzzy relational models
- Fuzzy decision trees
- Fuzzy neural networks

The processes of verification and validation are concerned with the fundamental issues of the development of the model and assessment of its usefulness. Verification is concerned with the analysis of the underlying processes of constructing the fuzzy model. On the other hand, validation is concerned with ensuring that the model meets the requirements of the customer. The performance index expressing the accuracy of a fuzzy model to a regression problem is to compute the distance between the numeric data and the numeric output of the model, cf. (Pedrycz and Gomide, Fuzzy Systems Engineering: toward human-centric computing 2007).

$$Q = \sum_{k=1}^{N} \left\| \hat{y}_k - \text{target}_k \right\|^2 \tag{2.12}$$

2.2.2. Interpretability of fuzzy models

In the highly diversified landscape of fuzzy modeling, there are several visible design tendencies to enhance the transparency (visibility) of fuzzy models. With the increasing accuracy of fuzzy models, which quite commonly happens at expense of their interpretability, there are a number of fundamental studies, which raise awareness about the need for the retention or enhancement of transparency of fuzzy models. One can refer here to a number of interesting studies documented in the literature, see (Alcala, et al. 2009) (Chen, Chen and Pedrycz 2010) (Eftekhari, et al. 2008) (Ishibuchi and Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning 2007). Some more recent studies carried out with this regard are reported in (Velez, et al. 2010).

The fundamental question being formulated in this context concerns a quantification of the notion of interpretability: what makes the fuzzy model more transparent? The answer is not straightforward as the problem itself is truly a multifaceted one. In rule-based fuzzy models (which is the commonly topology encountered in the area), we can think of several main design alternatives, which are also already quite visible in the literature:

(a) *minimization of the number of rules.* The number of rules directly impacts the interpretability of the resulting fuzzy model. Retaining all variables, the number of rules relates directly to the number of fuzzy sets formed for each variable, so a possible reduction of the total number of rules links to the lower number of information granules. This could be realized to some extent not sacrificing accuracy by optimizing a distribution of these fuzzy sets in the respective input spaces (Roubos and Setnes 2001).

(b) *hierarchical organization of rule-based models.* Hierarchies of fuzzy models, and fuzzy rules, in particular, could be advantageous in the reduction of the overall number of rules and thus contributing to the interpretability of the fuzzy model. While the multilevel structures of models are beneficial, there is an open question of interpretation of intermediate constructs emerging at the higher levels of the hierarchy.

(c) *reduction of dimensionality of the input space*. Bearing in mind that the number of rules directly gives rise to the combinatorial explosion (curse of dimensionality), it becomes critical to reduce the number of input variables. Dimensionality reduction has been around in data analysis (e.g., present in the form of data transformations such as those realized by linear or nonlinear projections), singular value decomposition (SVD) techniques and alike. This trend is also visible in the realm of fuzzy modeling, see e.g., (Roubos and Setnes 2001). From the perspective of interpretability of fuzzy models, it is essential that rather than realizing transformations of the original variables (resulting in new, non-interpretable variables), one concentrates on the most essential (meaningful) set of the variables by discarding less essential ones.

(d) *forming tradeoffs between explicit-implicit treatment of input variables*. There has been an overwhelming tendency to treat the input variables individually in the condition part of the rules of the model. While in this way, the interpretability has been retained, the dimensionality of the model started to grow quite rapidly. The use of fuzzy clustering naturally gives rise to rules whose conditions are not formed for the individual inputs but for all inputs. For instance, in Fuzzy C-Means, the obtained information granules – clusters capture implicitly all variables in the form of fuzzy sets defined as consecutive rows of the partition matrix. So the rules read as

$$\text{-if input } \mathbf{x}\ (= [x_1, x_2, \ldots, x_n]^T) \text{ is } R_i \text{ then } y = f_i(\mathbf{x}, \mathbf{a}_i) \qquad (2.13)$$

where i = 1, 2, …, c with “c” being the number of clusters. One can acknowledge that while the number of rules (c) is quite low, the interpretability at the level of individual input variables is not present in the direct way. A certain way of enhancing the interpretability aspect has been discussed and the information granules have been decomposed into the Cartesian product of the individual fuzzy sets and a relational (multidimensional) “remainder”, that is

$$R_i \approx A_i \times B_i \times C_i \times R_i^{\sim} \qquad (2.14)$$

where A_i, B_i, C_i … are fuzzy sets defined in the successive input variables and $R_i^{\sim}$ is the “remainder” defined over the Cartesian product of the remaining variables. In this way, the interpretability of the resulting rules of the form

-if input x_1 is A_i and x_2 is B_i and x_3 is C_i and $\mathbf{x}^{\sim}$ is $R^{\sim}_i$ then $y = f_i(\mathbf{x}, \mathbf{a}_i)$ (2.15)

is enhanced as a number of variables (information granules) show up in a direct fashion. The choice of these variables themselves could be a subject of a separate optimization procedure.

(e) *accommodation and reliance on logic aspects of the underlying topologies of fuzzy models*. The underlying direction is to construct fuzzy models based on a logic-based oriented blueprint (Gobi and Pedrycz 2007) (Liang and Pedrycz 2009). This concerns both the use of logic operators encountered in fuzzy sets (t-norms and t-conorms) while endowing them with parametric flexibility. The constructs of fuzzy neurons and logic processors are representative examples of this tendency of fuzzy modeling. Along with the learning capabilities, their transparency helps translate such networks into a collection of confidence-quantifiable set of rules. Another alternative exploited in this area is to proceed with Boolean (two-valued) logic models (for which there is a plethora of simplification techniques present in digital systems design) and use their reduced, simplified versions as a blueprint (skeleton) over which a fuzzy model can be formed.

The investigations put forward in our following study are positioned in the main stream of the design alternatives where the dimensionality of the input space can be reduced by eliminating some less essential input variables. One can stress here that this process of reduction is in line with the increased interpretability through shortening the length of condition parts in the rules. It is also worth noting that some other techniques enhancing interpretability could be applied to the reduced input space obtained in this manner. The granulation process comes with some additional flexibility in terms of the adjustable shape of membership function controlled by the fuzzification coefficient.

2.2.3. Functional (Local) fuzzy models

Functional fuzzy models are one of the most popular types of rule-based fuzzy models. Now let's briefly review the develop process. Assume that we have a number of functional fuzzy models (rules) at hand. The function in consequent part is linear. A linear model f(**x**, **a**$_i$) is constructed in a straightforward way commonly reported in the literature as the optimization problem can be handled analytically. With the squared error treated as the underlying performance index, the optimal coefficients of the model are derived in a standard fashion. Let us rewrite the model in an explicit way by using the activation levels of the individual rules. For the i-th local model we have

$$f_i(\mathbf{x},\mathbf{a}_i) = a_0 + a_1^i x_1 + a_2^i x_2 + \ldots + a_n^i x_n = \sum_{j=1}^{n} a_j^i x_j + a_0 \qquad (2.16)$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n, 1]^T$. The aggregation of the local models is realized in the form

$$\hat{y}_k = \sum_{i=1}^{c} A_{ik} f_i(\mathbf{x}_k, \mathbf{a}_i) \qquad (2.17)$$

where we use a shorthand notation λ_{ik} to denote $A_{ik} = u_i^m(\mathbf{x}_k) / \sum_{i=1}^{c} u_i^m(\mathbf{x}_k)$, with $u_i(\mathbf{x}_k)$ being the ik-th element of the partition matrix.

We optimize the structure of this fuzzy model by minimizing the squared error of the differences between the output of the model and the data

$$Q = \frac{1}{N}\sum_{k=1}^{N}(\hat{y}_k - \text{target}_k)^2 = \frac{1}{N}\sum_{k=1}^{N}\left(\sum \lambda_{ik}(a_0 + a_1x_1 + a_2x_2 + \ldots + a_nx_n) - \text{target}_k\right)^2 = \frac{1}{N}\sum_{k=1}^{N}\left(\sum_{i=1}^{c} \mathbf{z}_{ik}^T \mathbf{a}_i - \text{target}_k\right)^2 \qquad (2.18)$$

The performance index can be rewritten as follows

$$\min_{\mathbf{a}} Q\ (\mathbf{a}) = \|\mathbf{target} - Z\mathbf{a}\|^2 \qquad (2.19)$$

where

$$Z = \begin{bmatrix} \mathbf{z}_{11}^T & \mathbf{z}_{21}^T & \cdots & \mathbf{z}_{c1}^T \\ \mathbf{z}_{12}^T & \mathbf{z}_{22}^T & \cdots & \mathbf{z}_{c2}^T \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{z}_{1N}^T & \mathbf{z}_{2N}^T & \cdots & \mathbf{z}_{cN}^T \end{bmatrix} \qquad (2.20)$$

$$\mathbf{z}_{ik} = \lambda_{ik}\mathbf{x}_k \qquad (2.21)$$

$$\|\mathbf{target} - Z\mathbf{a}\|^2 = (\mathbf{target} - Z\mathbf{a})^T(\mathbf{target} - Z\mathbf{a}) \qquad (2.22)$$

The optimal solution is expressed in the form

$$\mathbf{a}_{opt} = Z^{\#}\text{target} \qquad (2.23)$$

$$Z^{\#} = (Z^T Z)^{-1} Z^T \qquad (2.24)$$

Note that higher order local models (say 2nd order polynomials) can be constructed in the same manner. When we build functional fuzzy models in the following chapters, we will use this result directly.

2.2.4. Artificial neural networks: historical overview

Artificial neural networks are systems motivated by the distributed, massively parallel computation in the brain that enables it to be so successful at complex control and recognition/classification tasks. The biological neural network that accomplishes this can be mathematically modeled (caricatured) by a weighted, directed graph of highly interconnected nodes (neurons). The artificial nodes have almost always simple activation functions whose arguments are the weighted summation of the inputs to the node; early work on neural networks and some current work uses node functions taking on only binary values. After a period of active development in the 1950's and 1960's, that slowed in the face of the limitations of the networks then being explored, neural networks experienced a renaissance in the 1980's with the work of Hopfield (Hopfield 1982) on the use of networks with feedback (graphs with cycles) as associative memories, and that of Rumelhart et al. (Rumelhart, Hinton and Williams, Learning representations by back-propagating errors 1986) (Rumelhart and McClelland, Parallel Distributed 1986) on backpropagation training and feedforward networks that could "learn" from input-output examples provided in a training set. Learning in this context is carried out by a descent-based algorithm that adjusts the network weights so that the network's response closely approximates the desired response specified by the training set. The capabilities of neural networks were quickly exploited in a great number of applications to pattern classification, control, and time-series forecasting.

Artificial neural networks are viewed as effective tools for solving problems such as pattern classification, speech recognition, image analysis, forecasting, and nonlinear or nonstationary system modeling. The main elements in the artificial network topology are neurons and connections (weights and biases). Each neuron includes two parts: sum of weighted inputs and an activation function. Feedforward neural networks (FNN's) and recurrent neural networks (RNN's) (Musilek, Pelikan, et al. 2006) are the two major classes of neural networks (NN's) widely used. Feedforward multilayered networks have been employed to realize complicated nonlinear decision/mapping functions. It has been shown that feedforward networks with one hidden layer can approximate any continuous mapping function, cf. (Blum and Li 1991) (Cybenco 1989) (Funahashi 1989) (Hornik, Stinchcombe and White 1989) (Marmarelis and Zhao 1997). An RNN consists of a large number of feedforward and feedback connections that exhibit complex dynamics. In some applications, an RNN with a smaller network size may be equivalent to rather complicated feedforward network architecture. Before the birth of these two neural networks, some early models like McCulloch-Pitts, Perceptron, ADALINE, et al contribute to the history of artificial neural networks' development.

2.2.5. Development of fuzzy models and neural networks in literature

Due to the important roles of fuzzy models in human-centric system modeling, and of artificial neural networks in approximating any nonlinear function, we decide to start our study in system modeling with granular architectures of CI by using these two models. First let's look at what has been done so far in literature about fuzzy models and neural networks.

Khoshgoftaar (Khoshgoftaar, et al. 1997) et al used neural networks to realize software quality modeling as they are adept at modeling nonlinear functional relationships that are difficult to model with other techniques. Iatrou et al (Iatrou, Berger and Marmarelis 1999) discussed a neural network being composed of a single hidden layer, polynomial activation functions and the backpropagation algorithm for the modeling of nonlinear Volterra systems. More studies on the development of neural networks with MLP topology please refer to (Ridella, Rovetta and Zunino 1997) (de Weerdt, Chu and Mulder 2009) (Wedge, et al. 2006) (Liu and Li 2004) (Ishibuchi and Nii, Improving the generalization ability of neural networks by interval arithmetic 1998) (Fang, et al. 2006).

Wang’s et al (Wang and Langari 1994) Sugeno-Takagi-Kang models win other modeling methods by the advantages of simplicity, flexibility, high accuracy in which the fuzzy discretization technique plays a central role in this approach and can be handled by an automatic procedure. Chen et al (Chen and Xi 1998) introduced an adaptive fuzzy inference system to cope with the problems often encountered in modeling nonlinear systems: conflict between overfitting and good generalization and low reliability. Jin (Jin 2000) tried to develop a fuzzy model of high-dimensional system in which many optimal strategies are used: optimal fuzzy rules covering extrema for initial fuzzy rule system, genetic algorithm and the gradient method for optimizing structure and parameters, low firing strength, fine training the fuzzy rules with regularization, and so on. Stylios et al (Stylios and Groumpos 2004) used fuzzy cognitive maps to model complex systems and applied their method to model and describe the behaviour of a heat exchanger system. Liu et al (Liu and Li 2005) used the probabilistic modeling method to improve the stochastic modeling capability and finally come out with a probabilistic fuzzy logic system. This system is applied to a function approximation problem and a robotic system.

To combine the fuzzy theory and neural networks, Lin et al (Lin and Cunningham 1995) introduced a simple fuzzy-neural network with fuzzy curves identifying significant input variables, determining model structure and setting the initial weights and proved that it can represent any continuous function. Farag' s et al (Farag, Quintana and Lambert-Torres 1998) approach combines the merits of the fuzzy logic theory, neural networks, and genetic algorithms to realize linguistic modeling of complex irregular systems with fuzzy-neuro network form which can handle both quantitative (numerical) and qualitative (linguistic) knowledge. Juang et al (Juang and Chen, A recurrent fuzzy-network-based inverse modeling method for a temperature system control 2007) designed a recurrent fuzzy network developed from a series of TSK-type fuzzy if-then rules in which the Kalman filter and gradient descent learning algorithms are used for parameter learning. It has been applied to a real water bath temperature control plant. Lin et al (Lin, et al. 2007) developed a frequency controlled three-phase induction generator system which uses fuzzy modeling to determine the flux control current and the maximum output power introduces online training recurrent fuzzy neural network with backpropagation algorithm to track controller of dc-link power. Juang et al (Juang and Hsieh, A locally recurrent fuzzy neural network with support vector regression for dynamic-system modeling 2010) proposed a recurrent model with five-layer recurrent network structure and the structure and parameter learning are realized by a one-pass clustering algorithm and an iterative linear SVR algorithm respectively. In Yeh' s et al (Yeh, Jeng and Lee 2011) model, a type-2 fuzzy Takagi-Sugeno-Kang IF-THEN rule is derived from each cluster to form a fuzzy rule base and a fuzzy neural network is constructed based on it. The parameters are refined by a hybrid learning algorithm which incorporates particle swarm optimization and a least squares estimation.

We can see a plethora of extension studies about neural networks and fuzzy modeling in literature; however, the collaboration of different sources of knowledge and generalization of system models remain challenging issues.

2.3. Analytic Hierarchy Process

Decision making, especially group decision making is very important issues in real problems which we are constantly faced with. Compared to most unconsciously decision making, logical decision making is an important part of all science-based professions. Decisions made collectively tend to be more effective than decisions made by a single individual.

Group decision making has many application fields in which various kinds of methods are explored to solve the problems. In business and management, analytic hierarchy process (AHP) is a widely used procedure for group decision making. It has been well demonstrated the effectiveness of AHP.

AHP developed by Thomas L. Saaty in 1970s is a structured technique based on mathematics and psychology for organizing and analyzing complex decisions, especially in group decision making (Saaty, Introduction to a modeling of social decision process 1983) (Saaty, The analytic hierarchy process, planning, priority setting, resource allocation 1980) (Saaty, Axiomatic foundation of the analytic hierarchy process 1986). AHP also forms an interesting alternative used to estimate a collection of numbers which can be translated into membership functions or preferences.

Basically, a system is an abstract model for a real-life structure such as the nervous system of a human, the government of a city, the transportation network of a state. In systems language we evaluate the impact of various components of a system on the entire system and find their priorities. For practical purposes a system is often regarded in terms of its "structure" and "function". A hierarchy is an abstraction of the structure of a system to study the functional interactions of its components and their impacts on the entire system. This abstraction can take several related forms, all of which essentially descend from an apex (an overall objective), down to sub-objectives, down further to forces which affect these sub-objectives, down to the people who influence these forces, down to the objectives of the people and then to their policies, still further down to the strategies, and finally, the outcomes which result from these strategies. The comparison of these outcomes invoke the born of AHP mechanisms.

2.3.1. The framework of AHP

Let us look at a related problem which has interesting applications. It is concerned with measurement. Suppose we are given a set of objects which are all sufficiently light and can be lifted by hand. In absence of a weighting instrument, we wish to estimate their relative weights. One way to realize this is to guess the weight of each object directly in pounds. Another method which utilizes more of the available information in the experiment is to compare the objects in pairs, by lifting one and then lifting another and back to the first and then again the second and so on until we have formulated a judgment as to the relative weight (ratio) of each pair of objects. The problem is then to adopt a meaningful scale for the pairwise comparisons.

Note that consistency in any kind of measurement cannot be taken for granted. Since perfect consistency is difficult to attain in practice, what we need is a way of evaluating how bad it is. The term consistency here not merely mean the traditional requirement of the transitivity of preferences (if apples are preferred to oranges and oranges are preferred to bananas, then apples must be preferred to bananas), but the actual intensity with which the preference is expressed transits through the sequence of objects in the comparison.

Many systems or sub-systems show a tendency of being formed in a structure of hierarchy. There are many advantages of this structure.

(1) A hierarchy can be used to describe how changes in priority at upper levels affect the priority of elements in lower levels.
(2) They give great detail of information on the structure and function of a system in the lower levels and provide an overview of the actors and their purposes in the upper levels.
(3) Natural systems assembled hierarchically.
(4) They are stable and flexible.

Let's use an example to show how to determine a priority scale. Assume A, B, C, D stand for apples, oranges, bananas and pears. We develop a priority scale of relative preference for these fruits. Judgements will be obtained from an individual. For example, "How do you prefer A to B?" He will then give one of the numbers for comparison described in the table below and this judgement will be entered in a matrix in position (A, B). By convention, the comparison of preference is always of an activity appearing in the column on the left against an activity appearing in the row on top. We then have the pairwise comparison matrix with four rows and four columns (a 4*4 matrix).

preference	A	B	C	D
A				
B				
C				
D				

Prior to making any assessment, the expert is provided with a finite scale with values spread in between 1 and 9. Some other alternatives of the scales such as those involving 5 or 7 levels could be sought as well. This depends on the different situation and requirements. If A is strongly preferred to B when being considered in the context that we mentioned above, then this judgment is expressed by assigning high values of the available scale, say 8 or 9 and inserted in the position (A, B) where the row of A meets the column of B. If we still sense that A is preferred to B yet the strength of this preference is lower in comparison with the previous case, then this is quantified using some intermediate values of the scale, say 5 or 6. If no difference is sensed, the values close to 1 are the preferred choice, say 2 or 1. The value of 1 indicates that A and B are equally preferred. On the contrary, if B is preferred to A, the corresponding entry assumes values below one. Given the reciprocal character of the assessment, once the preference of A to B has been quantified, the inverse of this number is plugged into the entry of the matrix that is located at the position (B, A). As indicated earlier, the elements on the main diagonal are equal to 1.

Let use return to our example. The diagonal elements in the matrix are all equal to 1. There are 12 elements left. According to the reciprocal principle, we have to determine 6 elements. The other 6 elements are the reverse comparisons. The individual provides all these elements.

Let's look at the details of ideal case--a consistent matrix case. Consider a collection of elements: z_1, z_2, ... z_n (those could be, for instance, some alternatives whose allocation to

preference	A	B	C	D
A	1	5	6	7
B	1/5	1	4	6

C	1/6	1/4	1	4
D	1/7	1/6	1/4	1

a certain fuzzy set is sought) for which their membership grades or preferences: $A(z_1)$, $A(z_2)$, ..., $A(z_n)$ are given. In order to compare the results, we assume that $A(z_i)$ are all normalized into the unit range [0, 1]. Then they are organized in a so-called reciprocal matrix of the following.

$$R = [r_{ij}] = \begin{bmatrix} \frac{A(z_1)}{A(z_1)} & \frac{A(z_1)}{A(z_2)} & \cdots & \frac{A(z_1)}{A(z_n)} \\ \frac{A(z_2)}{A(z_1)} & \frac{A(z_2)}{A(z_2)} & \cdots & \frac{A(z_2)}{A(z_n)} \\ & \cdots & \cdots & \\ \frac{A(z_n)}{A(z_1)} & \frac{A(z_n)}{A(z_2)} & \cdots & \frac{A(z_n)}{A(z_n)} \end{bmatrix} = \begin{bmatrix} 1 & \frac{A(z_1)}{A(z_2)} & \cdots & \frac{A(z_1)}{A(z_n)} \\ \frac{A(z_2)}{A(z_1)} & 1 & \cdots & \frac{A(z_2)}{A(z_n)} \\ & \cdots & \cdots & \\ \frac{A(z_n)}{A(z_1)} & \frac{A(z_n)}{A(z_2)} & \cdots & 1 \end{bmatrix} \qquad (2.25)$$

Noticeably, the diagonal values of R are equal to 1. The entries that are symmetrically positioned with respect to the diagonal satisfy the condition of reciprocality, that is, $r_{ij} = 1/r_{ji}$. Furthermore, an important transitivity property holds, that is, $r_{ik}r_{kj} = r_{ij}$, for all indexes i, j and k. This property holds because of the way in which the matrix has been constructed. Let us now multiply the reciprocal matrix by the vector of elements of $A = [A(z_1), A(z_2), ..., A(z_n)]$. For the ith row of R (which is the ith entry of the resulting vector of results) we obtain

$$[RA]_i = \begin{bmatrix} \frac{A(z_i)}{A(z_1)} & \frac{A(z_i)}{A(z_2)} & \cdots & \frac{A(z_i)}{A(z_n)} \end{bmatrix} \begin{bmatrix} A(z_1) \\ A(z_2) \\ \cdots \\ A(z_n) \end{bmatrix} \qquad (2.26)$$

where i = 1, 2, ..., n. Thus the ith element of the vector is equal to $nA(z_i)$. Overall once completing the calculations for all i, this leads us to the expression RA = nA. In other words, we conclude that A is the eigenvector of R associated with the largest eigenvalue of R that is equal to n. In the above scenario, we have assumed that the values $A(z_i)$ are given and then showed what form of results could they lead to. In practice the values of $A(z_i)$ are not given and have to be looked for.

When we have a reciprocal matrix at hand, the next step is to calculate priorities and the eigenvalue of the matrix. The principle eigenvector is the one with the largest eigenvalue and when it is normalized, the vector becomes of priorities. The normalized version of the eigenvector is then the membership function of the fuzzy set we considered when doing all pairwise assessments of the elements of its universe of discourse. The pairwise evaluations are far more convenient and manageable in comparison to any effort we make when assigning membership grades to all elements of the universe in a single step. Practically, the pairwise comparison helps the expert focus only on two elements once at a time thus reducing uncertainty and hesitation while leading to the higher level of consistency. The assessments are not free of bias and could exhibit some inconsistent evaluations. In particular, we cannot expect that the transitivity requirement could be fully satisfied. Fortunately, the lack of consistency could be quantified and monitored. The largest eigenvalue is denoted as λ_{max} and the corresponding eigenvector is **e**. λ_{max} computed for R is always greater than the dimensionality of the reciprocal matrix (recall that in reciprocal matrices the elements positioned symmetrically along the main diagonal are inverse of each other), $\lambda_{max} > n$ where the equality $\lambda_{max} = n$ occurs only if the results are fully consistent. The ratio

$$\nu = (\lambda_{max} - n) / (n-1) \qquad (2.27)$$

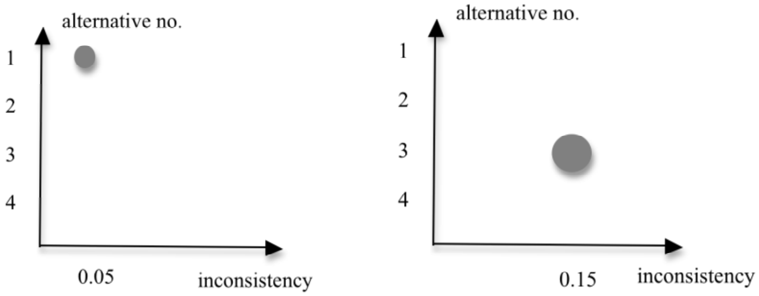
can be regarded as an index of inconsistency of the data; the higher its value, the less consistent are the collected experimental results. This expression also can be sought as the indicator of the quality of the pairwise assessments provided by the expert. If the value of ν is too high exceeding a certain superimposed threshold, the experiment may need to be repeated. Typically, if ν is less than 0.1 or 0.2 (this threshold differs for different sizes of matrices) (Apostolou and Hassell 2002) (Chu and Liu 2002), the assessment is sought to be consistent whereas higher values of ν call for a re-examination of the experimental data and a rerun of the experiment. It is worth noting that any value of threshold needs to be cast in a context of a certain application; refer also to an interesting discussion on this issue covered in (Apostolou and Hassell 2002) (Chu and Liu 2002).

2.3.2. AHP in group decision making

AHP developed in a series of studies by Saaty, cf. (Saaty and Ozdemir, Negative priorities in the analytic hierarchy process 2003) (Saaty, How to handle dependence with the analytic hierarchy process 1987) (Saaty and Hu, Ranking by eigenvector versus other methods in the analytic hierarchy process 1998) (Saaty, Introduction to a modeling of social decision process 1983) (Saaty, A new macroeconomic forecasting and policy evaluation method using the analytic hierarchy process 1987) and further researched and generalized by others (Tung and Tang 1998) is aimed at forming a vector of preferences given a finite set of alternatives. Once the reciprocal matrix has been completed through running a series of pairwise comparisons, the maximal eigenvalue λ_{max} and the associated eigenvector of R, say **e**, are determined. This eigenvector (normalized or not) serves as a vector of preferences of the alternatives.

One of the main advantages of the AHP method is that it becomes far more convenient to evaluate only two alternatives at a time, especially in case of a large number of alternatives and this is a way in which the AHP method supports the evaluations. Another advantage comes with an ability to assess the consistency of the pairwise comparisons. In an ideal case, the largest eigenvalue is equal to the number of alternatives. When we encounter some level of inconsistency (which is a reflection of some violation of the transitivity property), the eigenvalue gets higher. Given this, a level of inconsistency of evaluations (the reciprocal matrix) is expressed in ν.

The results of the AHP can be conveniently captured and visualized in the form of a so-called decision profile. It offers a convincing view at the results of rating the alternatives and the associated quality of the process. We plot the result - the best alternative is expressed in the coordinates of the inconsistency level (ν) and the number (index) of the alternative. In addition, the size of the symbol (here circle) is made proportional to the difference between the membership of the best alternative and the next one appearing in the increasing order of preference. Two illustrative examples are presented in Figure 5.



(a) (b)

Figure 5. Examples of decision profiles: (a) e = [1.0 0.8 0.9 0.2] ν =0.05, and (b) e=[0.3 0.5 1.0 0.6] and ν =0.15. The radii of the circles are equal to 0.1 (=1-0.9) and 0.4 (=1-0.6), respectively.

Recently, more and more studies show the outstanding performance of granular computing in decision making problems, cf. (Cakir 2008) (Dong, Hong, et al. 2011) (Herrera, Herrera-Viedma and Verdegay, A rational consensus model in group decision making using linguistic assessments 1997) (Herrera-Viedma, Martinez, et al. 2005) (Herrera-Viedma, Alonso, et al. 2007) (Kangmao, Wang and Chun 2005) (Mata, Martinez and Herrera-Viedma 2009) (Wang and Chen 2008). However, the studies reported so far exploit more or less sophisticated ways of adjusting/transforming initial judgements (preferences) of individuals. It remains a challenging issue to obtain a decision with high level of consensus.

2.4. Particle Swarm Optimization and Genetic Algorithms

Modern heuristic optimization techniques have aroused great interest among the scientific community in a wide variety of fields for the last years, because of their ability to solve multimodal and multidimensional discontinuous problems in a near optimal manner. Basically, these techniques try to mimic natural, biological or even cultural evolution, depending on the nature of the process. The most representative algorithms include simulated annealing (SA) (Kirpatrick, Gelatt and Vecchi 1983), genetic algorithms (GA) (Goldberg 1989), particle swarm optimization (PSO) (Kennedy and Eberhart, Particle swarm optimization 1995), ant colony optimization (ACO) (Dorigo, Maniezzo and Colorni 1996). Compared to other algorithms, PSO is easier to understand, easier to implement, requiring fewer lines of code, and having a smaller number of parameters to be tuned. Genetic algorithms motivated by Darwin's theories of evolution and the concept of "survival of the fittest" have also been widely applied as they have a profound foundation. In this section, the evolutionary optimization of an objective function and two-objective function through particle swarm optimization and genetic algorithm is described.

Particle swarm optimization is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 (Kennedy and Eberhart, Particle swarm optimization 1995). It uses a simple mechanism that mimics swarm behaviour in birds flocking and fish schooling to guide the particles to search for globally optimal solutions. As PSO is easy to implement, it has rapidly progressed in recent years and with many successful applications seen in solving real-world optimization problems. Though it is a comparable new optimization tool, there have been many developments, applications and resources since its origin.

Genetic algorithms use processes analogous to genetic recombination and mutation to promote the evolution of a population that best satisfies a predefined goal. In the selective breeding or crossover process, fit individuals are chosen to produce more offspring than less fit individuals, which tends to homogenize the population and improve the average result as the algorithm progresses. Subsequent mutations of the offspring add diversity to the population and explore new areas of the parameter search space. The algorithm starts by generating an initial population of random candidate solutions. Each individual in the population is then awarded a score based on its performance. The individuals with the best scores are chosen to be parents, which are cut and spliced together to make children as illustrated in the breakout. To add some diversity to the population, some random mutations are applied. These children are scored, with the best performers likely to be parents in the next generation. At some point the process is terminated and the best scoring individual in the population is taken as the final answer.

In our experiments, we mainly use PSO algorithm. We first present some improved PSO algorithms and pseudo codes for selected algorithms. The improved PSO algorithms focus on mechanisms for updating a particle's velocity, selection of population topology, the number of particles, and the strategies for updating at run time various parameters that influence performance. Then multiple-objective PSO is introduced with pseudo codes.

2.4.1. A single-objective PSO

A. The original PSO algorithm

To optimize an n-dimensional continuous objective function f: $\mathbf{R}^n \rightarrow R$, a population of particles P = {$\mathbf{p}_1, \ldots, \mathbf{p}_N$} (called swarm) is randomly initialized in the solution space. The objective function determines the quality of the solution represented by a particle's position. We assume the following discussion to minimization problems.

At any time step t, a particle $\mathbf{p}_i$ has an associated position vector $\mathbf{s}_i = [s_{ij}]$, j = 1, 2, …, n and a velocity vector $\mathbf{v}_i = [v_{ij}]$. A vector $\mathbf{pb}_i = [pb_{ij}]$ stores the best position the particle has ever visited. The best vector $\mathbf{pb}_i$ in a particle's neighbourhood (called local best) is a vector $\mathbf{pg} = [pg_j]$.

PSO algorithms update the particles' velocities and positions iteratively until a stopping criterion is met. The basic velocity- and position-update rules are

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(pb_{ij}(t)\text{-}s_{ij}(t)) + c_2 r_{2j}(pg_j(t)\text{-}s_{ij}(t)) \quad (2.28)$$

$$s_{ij}(t{+}1) = s_{ij}(t) + v_{ij}(t{+}1) \quad (2.29)$$

where t is the current generation, c_1 and c_2 are two constants acceleration coefficients being typically set to 2. In the following chapters, if there is no specific explanation, we use this setting. r_{1j} and r_{2j} are two uniformly distributed random numbers independently generated within [0, 1] for the jth dimension. There are also some limits imposed on the velocities, say $\mathbf{v}_{max}$ and $\mathbf{v}_{min}$. Their values are related to the scope of the search spaces and prevent velocities from growing to extremely values.

The searching loop will stop when some conditions are met or the maximum iterations reaches.

B. Time-Varying Inertia Weight Particle Swarm Optimizers

Shi and Eberhart (Shi and Eberhart, A modified particle swarm optimizer 1998) (Shi and Eberhart, Empirical study of particle swarm optimization 1999) noticed that the first term of the right-hand side of velocity update formula plays the role of a particle's "inertia" and they introduced the idea of an inertia weight. The velocity update rule was modified to

$$v_{ij}(t+1) = \eta(t)v_{ij}(t) + c_1 r_{1j}(pb_{ij}(t)\text{-}s_{ij}(t)) + c_2 r_{2j}(pg_j(t)\text{-}s_{ij}(t)) \quad (2.30)$$

where η(t) is the time-dependent inertia weight. Shi and Eberhart proposed to set the inertia weight according to a time-decreasing function so as to have an algorithm that initially explores the search space and only later focuses on the most promising regions. The function used to schedule the inertia weight is defined as

$$\eta(t) = \frac{t_{\eta max} - t}{t_{\eta max}}(\eta_{max} - \eta_{min}) + \eta_{min} \quad (2.31)$$

where $t_{\eta max}$ marks the time at which $\eta(t) = \eta_{min}$, η_{max} and η_{min} are the maximum and minimum values the inertia weight can take, respectively. Normally, $t_{\eta max}$

coincides with the maximum time allocated for the optimization process. We identify this variant as decreasing-IW PSO.

Experimental results showed that this approach is effective (Shi and Eberhart, Parameter selection in particle swarm optimization 1998) (Shi and Eberhart, A modified particle swarm optimizer 1998) (Shi and Eberhart, Empirical study of particle swarm optimization 1999). The pseudo code of the decreasing-IW PSO is presented as follows

```
Create the initial population of swarm
t=0;
While t<maximum number of generations
    Update inertia weight using (2.31)
    t=t+1
    For each particle i do
         Calculate the optimization criterion, namely $q_1(\mathbf{s}_i(t))$
          If $q_1(\mathbf{s}_i(t))<q_1(\mathbf{s}_i(t-1))$
            Replace $\mathbf{s}_i(t-1)$ with $\mathbf{s}_i(t)$
          Endif
    Endfor
    Find global best--the particle with largest $q_1$
    Save the position of this particle and its $q_1$
    For each particle i do
         Update the velocity $\mathbf{v}_i(t)$ according to (2.30)
          If $\mathbf{v}_i(t)$ satisfies the limitation of velocity
               Update the position $\mathbf{s}_i(t)$ by (2.29)
          Endif
    Endfor
Endwhile
```

C. Constricted Particle Swarm Optimizer

Clerc and Kennedy (Clerc and Kennedy 2002) added a constriction factor to the particles' velocity-update rule to avoid the unlimited growth of the particles' velocity. Equation is modified to

$$v_{ij}(t+1)=\chi(v_{ij}(t)+c_1r_{1j}(pb_{ij}(t)\text{-}s_{ij}(t))+c_2r_{2j}(pg_j(t)\text{-}s_{ij}(t))) \qquad (2.32)$$

where $\chi = 2/\left|2-c-\sqrt{c^2-4c}\right|$, c = c_1 + c_2, and c >4. Generally, c_1 and c_2 are set to 2.05, giving as result of χ equal to 0.729.

D. Fully Informed Particle Swarm Optimizer

Mendes et al. (Mendes, Kennedy and Neves 2004) proposed the fully informed particle swarm (FIPS), in which a particle uses information from all its topological neighbours. Clerc and Kennedy's constriction factor is also adopted in

FIPS; however, the value c (i.e., the sum of the acceleration coefficients) is equally distributed among all the neighbours of a particle.

For a given particle p_i, c is decomposed as $c_k = c/|N_i|$, $\forall p_k \in N_i$, N_i is the topology neighbourhood of p_i. The velocity-update equation becomes

$$v_{ij}(t+1) = \chi(v_{ij}(t) + \sum_{p_k \in N_i} c_k r_{kj}(pb_{kj} - s_{ij})) \qquad (2.33)$$

E. Self-Organizing Hierarchical Particle Swarm Optimizer with Time-varying Acceleration Coefficients

Ratnaweera et al. (Ratnaweera, Halgamuge and Waston 2004) proposed the self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients (HPSOTVAC), in which the inertia term in the velocity-update rule is eliminated. Additionally, if any component of a particle's velocity vector becomes zero (or very close to zero), it is reinitialized to a value proportional to v_{maxi}, which is the maximum velocity of the ith dimension allowed. This gives the algorithm a local search behaviour that is amplified by linearly adapting the value of the acceleration coefficients c_1 and c_2. The coefficient c_1 is decreased from 2.5 to 0.5, and the coefficient c_2 is increased from 0.5 to 2.5. In HPSOTVAC, the maximum velocity is linearly decreased during a run so as to reach 1/10 of its value at the end. A low reinitialized velocity near the end of the run allows particles to move slowly near the best region they have found. The resulting PSO variant is a kind of local search algorithm with occasional magnitude-decreasing unidimensional restarts.

F. Adaptive Hierarchical Particle Swarm Optimizer

The adaptive hierarchical PSO (AHPSO) (Janson and Middendorf 2005) modifies the neighbourhood topology at run time. It uses a tree-like topology structure in which particles with better objective function evaluations are located in the upper nodes of the tree. At each iteration, a child particle updates its velocity considering its own previous best performance and the previous best performance of its parent. Before the velocity-update process takes place, the previous best fitness value of any particle is compared with that of its parent. If it is better, the child and its parent swap their positions in the hierarchy. Additionally, AHPSO adapts the branching degree of the tree while solving a problem to balance the exploration-exploitation behaviour of the algorithm: a hierarchy with a low branching degree has a more exploratory behaviour than a hierarchy with a high branching degree. In AHPSO, the branching degree is decreased by degrees (one at a time) until a certain minimum degree is reached. This process takes place every number of iterations.

2.4.2. Multiple-objective PSO

The basic PSO algorithm has been extended to accomodate the problem formulation of multiobjective optimization problems (MOP), which is to search for a well extended, uniformly distributed, and near-optimal Pareto front. PSO is of particular interest in the context of multiobjective optimization considering its rapid convergence and a relatively low computing overhead. Here we review some well-known related works of MOPSO algorithms and give pseudo codes of one MOPSO.

Coello Coello et al. (Coello Coello, Pulido and Lechuga, Handling multiple objectives with particle swarm optimization 2004) proposed an approach which includes the following three features: incorporating Pareto dominance to handle problems with several objective functions, using a secondary (i.e., external) repository of particles to guide particles flight, incorporating a special mutation operator to improve the exploration capability. Leong et al. (Leong and Yen 2008) proposes an integration of a dynamic population strategy within the multiple-swarm MOPSO and designs adaptive local archives to improve the diversity within each swarm so that the algorithm obtains a nondominated front that is close to the true Pareto front and maintains the diversity of the solutions along the resulting Pareto front. It incorporates four strategies: 1) cell-based rank density estimation scheme to keep track of the rank and density values of the particles; 2) population-growing strategy to increase the population size to promote exploration capability; 3) population declining strategy to prevent the population size from growing excessively; and 4) adaptive local archives designed to improve the distributed solutions along the sections of the Pareto front that associate with each sub-swarm.

Other works about improvement and modification of PSO to MOPSO focus on different aspects of MOPSO. In these areas, selection and update mechanisms of best global leaders within the population is a basic but very important issue. Hu and Eberhart (Hu and Eberhart 2002) proposed a dynamic neighbourhood PSO that includes the following three criteria: dynamic neighbours, a new global best particle updating strategy, and one dimension optimization. The scheme of selecting and updating the leaders involves the particles' neighbourhood. At every iteration, each particle is assigned to the closest neighbourhood. Among the solutions in the neighbourhood, each particle finds the local best particle, as the global best and the local best are updated only when these solutions dominate it. This algorithm is only capable of dealing with a small number of objective functions. Another work by Zhang et al. (Zhang, et al. 2003) has suggested a selection scheme for global and local leaders that involves computing the new leaders via the proposed equation, which depends upon each objective function and the current iteration, and deciding whether the particles should follow their leaders based upon the proposed criteria.

It is difficult to control the velocity of each particle to perform its optimal flight in a high-dimensional search space in MOPSO. Hence, external archive is usually used to record any good particles found at each iteration. Mostaghim and Teich

(Mostaghim and Teich 2003) adopted the ε-dominance method to control the archive size and help reduce computational cost. Li (Li, A non-dominated sorting particle swarm optimizer for multiobjective optimization 2003) proposed a nondominated sorting particle swarm optimizer (NSPSO), in which the design elements and archive maintenance of NSGA-II are adopted in PSO design. Another prominent work was contributed by Coello Coello and Lechuga (Coello Coello and Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization 2002). They proposed a MOPSO algorithm that incorporates the concept of Pareto dominance and adopts archive controller, which stores and decides the membership of new nondominated solutions found at each iteration. In addition, an adaptive grid feature based upon the objective function values of archive members is applied to the archive, with the goal of producing a well-distributed Pareto front.

Typically, we consider two objectives, which are often in conflict. The results give rise to a Pareto front forming a collection of non-dominated vectors. To construct the front, we use the approach presented in (Coello Coello, Pulido and Lechuga, Handling multiple objectives with particle swarm optimization 2004).

At each generation, we calculate the values of the two objectives for each particle of the swarm, namely q_1 and q_2 (assume that their values are within [0, 1]). These two values form a two-dimensional vector. Then we find non-dominated vectors in the entire population and compare them with regard to the external repository. At the beginning of the process, this repository is empty. If the external repository is empty, then the current solution is accepted and stored there. Otherwise, we compare the new solution with the existing non-dominated vectors present in the external repository. If this new solution is dominated by an individual within the external archive, then such solution is discarded. If none of the elements in the external archive dominates the new solution, then such solution is stored there. If there are solutions in the archive that are dominated by the new solution, then such solutions are removed from the archive. Finally, the elements of the external repository are used to construct the Pareto front. To visualize the Pareto front, we display all solutions present in the external archive in a two-dimensional space with q_1- (1- q_2) coordinates.

To allow a quantitative assessment of the performance of the two-objective optimization algorithm and single-objective optimization algorithm, one metric is computed for the generated Pareto front: the average distance (Euclidean distance) between points on the Pareto front and the origin.

A pseudocode of the two-objective PSO is presented in the following way:

```
Create and initialize the swarm
Initialize the external repository as set it empty
t=0;
While t<maximum number of generations
    Update inertia weight using (2.31)
    t=t+1
    For each particle i do
        Calculate the optimization criteria, namely $q_1(\mathbf{s}_i(t))$ and $q_2(\mathbf{s}_i(t))$
        If $q_1(\mathbf{s}_i(t)) < q_1(\mathbf{s}_i(t-1))$ & $q_2(\mathbf{s}_i(t)) < q_2(\mathbf{s}_i(t-1))$
            Replace $\mathbf{s}_i(t-1)$ with $\mathbf{s}_i(t)$
        Endif
    Endfor
    Find all non-dominated vectors $[q_1, q_2]$ from the population and store them and their positions in two different archives R and S respectively, of a certain size.
    Calculate distance between each element in R and the original point and save them in $Q_e$
    Sort $Q_e$ in descending order and select top 20% of elements and denote it as $Q_{eo}$
    Randomly select a value from $Q_{eo}$ and its corresponding position as $\mathbf{pg}$
    Update velocities and positions of all particles according to (2.30), (2.29)
Endwhile
```

3. The Principle of Justifiable Granularity

Information granules regarded as conceptually justifiable constructs constitute building blocks of intelligent systems. They are used in various pursuits of system analysis and design. In this chapter, we introduce the principle of justifiable granularity that is used to form a meaningful representation of a collection of numeric values and how it is applied in a realistic problem especially when we construct a granular model. Then we fix the problem of a mapping of linguistic terms to information granules in which linguistic information is made operational based on intervals and fuzzy sets.

3.1. Motivation and Definition

The scientific theories and their application in industry have elevated accuracy of modeling of systems to a very high level. However, the vast amount of numeric data that become available nowadays in various domains of human beings highlight the fact that there is an urgent need for a more human-like information processing. We can name this activity as generalization or abstraction. In 1979, Zadeh first gave the term information granulation and emphasized the fact that the plethora of detail data does not amount to knowledge (Zadeh, Fuzzy sets and information granularity, in: M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.) 1979). Another important issue in system analysis is to make the process of knowledge building automated. Information granules are inherently associated with these two issues.

Most raw experimental data collected by human beings or machines are numeric. Granulation of information happens in the process of data organization and data comprehension. An interesting thing is that people granulate information in a subconscious manner. And this action is demonstrated making the ensuing cognitive process more effective.

So this chapter will introduce the basic framework that facilitates transition from numbers to semantically richer information granules. The principle of justifiable granularity is proposed with intent to having an effective and parametrically flexible vehicle to form information granules.

3.2. A Design of Granules

The principle of justifiable granularity is concerned with formation of a meaningful representation of a collection of numeric values (real numbers). Two representative categories of problems involve processing one or two-dimensional signals. In our study, we give an example of one dimension case, say representing numbers $\{d_1, d_2, \ldots, d_N\}$. Such a representation comes in the form of a certain information granule rather than a single numeric entity, no matter how such a single individual has been selected. For example, what is being done in statistics is realized in the language of probabilistic information granules. A sample of numeric data is represented not only by its mean or median (which is a very rough description) but also by the standard deviation. Both the mean and the standard deviation imply a realization of a certain probabilistic information granule, such as e.g., a Gaussian one. The probabilistic information granules are just one of the possibilities to build an information granule to represent a collection of numeric data. Some other formal approaches that could be engaged here involve sets, intervals, fuzzy sets, rough sets and others. Formally, we can view the process of granulation as a transformation $\mathcal{G}$ operating on $\{d_1, d_2, \ldots, d_N\}$ resulting in some information granule

$$\mathcal{G} : \mathbf{R} \rightarrow \boldsymbol{G}(\mathbf{R}) \qquad (3.1)$$

where $\boldsymbol{G}(\mathbf{R})$ is a granular construct, for instance, interval, $\boldsymbol{G}(\mathbf{R}) = \boldsymbol{P}(\mathbf{R})$ or fuzzy set, $\boldsymbol{G}(\mathbf{R}) = \boldsymbol{F}(\mathbf{R})$, etc.

Generally, in the formalism of information granules, the development of the corresponding granules is guided by a certain optimization criterion, which becomes a crux of the principle of justifiable granularity. Depending on what form of the information granules, we may have different optimization processes and requirements. In the following, we focus on interval form and illustrate the whole development process.

We discuss several general cases to venture into more algorithmic details of the realization of information granules. We show a construction of interval-based information granules.

3.2.1. Design of interval-based information granules

When we consider interval forms $\boldsymbol{G}(\mathbf{R}) = \boldsymbol{P}(\mathbf{R})$, there are two situations of formation according to the information of numeric data.

(a) The design of interval-based information granule of numeric data. The data are illustrated in Figure 6.

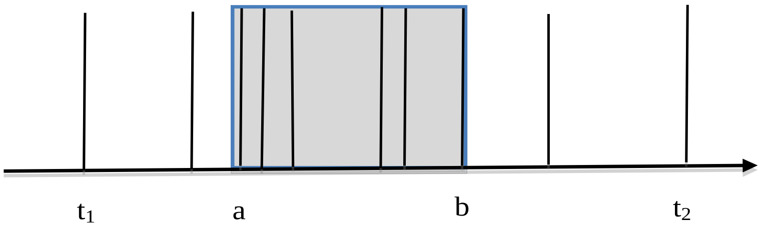


Figure 6. Realization of the principle of justifiable granularity for numeric data and interval form of information granules

In the optimization criteria, we manage two conflicting requirements. The first one is about forming an information granule of sufficiently high level of experimental evidence that is accumulated behind it and in this way supports its existence (and usage). The second one is about maintaining high specificity of the resulting information granule. In what follows, we show a construction of interval-based information granules in presence of some numeric evidence, see Figure 6. Consider two intervals: where **Ω** = [a, b], and Ψ = [t_1, t_2]; note that t_1 and t_2 are just the minimal and maximal elements in the data under consideration. Given the two conflicting requirements identified above when forming information granules, we set up the numeric interval **Ω** in such a way that (i) the numeric evidence accumulated within the bounds of **Ω** becomes as high as possible. One simple way to quantify this requirement is by counting the number of data falling within the bounds of the information granule (interval) **Ω**, that is card {$d_k \in$ **Ω**}, which has to be maximized. At the same time, we require that (ii) the support of **Ω** is as low as possible, which makes **Ω** specific enough. A certain way to combine these two conflicting requirements into a single criterion is to consider the ratio

$$Q = \frac{\text{card}(d_k \in \Omega)}{\text{supp}(\Omega)} \tag{3.2}$$

which is maximized with regard to the end-points of the interval, namely max Q.

(b) Here we design the interval information granule considering that the numeric data come with membership values; that is we are concerned with the pairs (d_k, μ_k) where μ_k stands for the k-th membership value. This design situation is illustrated in Figure 7.

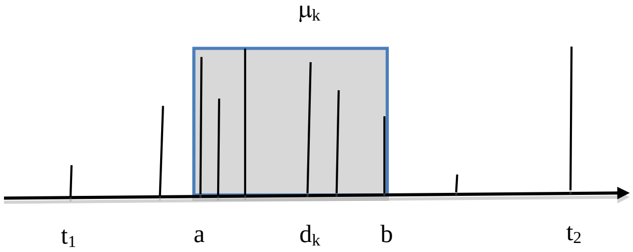


Figure 7. Realization of the principle of justifiable granularity for numeric data with membership -graded (weighted numeric data) and interval form of information granule and the optimization criterion

The same design development as discussed in (a) applies here. As each data point comes with the associated membership value, the numeric evidence accumulated within Ω has to be computed in such a way that they are present in the calculations and contribute to the accumulated experimental evidence behind Ω.

We determine the σ-sum of the evidence, that is $\sum_{d_k \in \Omega} \mu_k$. This leads to the maximization of the following performance index

$$Q = \frac{\sum_{d_k \in \Omega} \mu_k}{\text{supp}(\Omega)} \tag{3.3}$$

Alternatively, we can focus on the formation of the information granule Ω, which leads to the minimum of changes of the membership grades of the corresponding data. To admit d_k with membership μ_k to Ω, we need to change (elevate) the membership grade and this change is equal to $1-\mu_k$. Similarly, if we exclude d_k from Ω, the corresponding change (suppress) in membership value is μ_k. See Figure 8. The criterion of interest is that of the sum of all possible changes made to the membership grades. We construct Ω in such a way that the changes in membership values are as low as possible. Formally, the performance index is expressed as

$$Q = \sum_{d_k \in \Omega} (1-\mu_k) + \sum_{d_k \notin \Omega} \mu_k \tag{3.4}$$

and its minimization leads to the interval-type of information granule,

$$\text{Min}_{\,a,b:\,a<b}\ Q \tag{3.5}$$

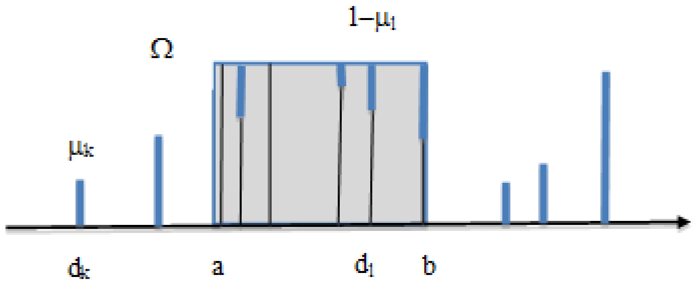


Figure 8. The design of interval information granule realized through the minimization of the criterion of modification of membership grades

3.2.2. Application of the principle on a family of numeric values

To illustrate the underlying idea, we use a simple example where we consider a set of following numeric data,

{5, 8, 10, 15, 20, 21, 26, 28, 34, 37}

An interesting alternative to design interval-based information granule of numeric data is:

$$Q = \frac{\text{card}(d_k \in \Omega)}{e^{\alpha|b-m|}} \tag{3.6}$$

where "b" represents the bound and "m" is the predefined numeric representative of the numeric data (for instance, mean, mode or median) which helps split the optimization process into two separate parts: we separately optimize the lower bound, "a" and the upper bound of the interval, "b". It is a general version of () offered some flexibility through the exponential function. Another case is to design the interval information granule considering that the numeric data come with membership values; that is we are concerned with the pairs (d_k, μ_k) where μ_k stands for the k-th membership value. The calculations involve the formula

$$Q = \frac{\sum_{d_k \in \Omega} \mu_k}{e^{\alpha|b-m|}} \tag{3.7}$$

Note that both (3.6) and (3.7) contain some adjustable parameter α which helps endow the method with some level of flexibility.

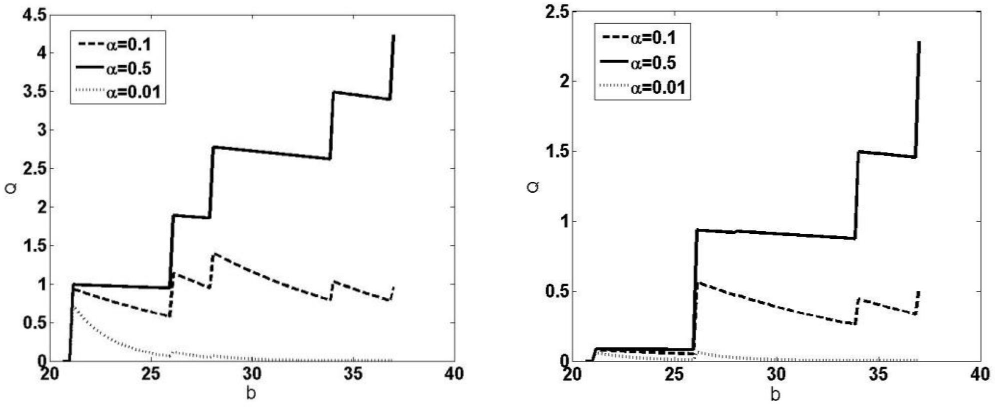
In real applications, numeric values may come with a membership values positioned in the unit range: [0, 1]. Thus, in the following study, we construct intervals by considering two aspects: (a) all membership values are equal to one (b) their membership values are different. For the given example above, assume the different memberships for the numeric values are as follows

[0.21, 0.99, 0.73, 0.66, 0.84, 0.09, 0.9, 0.01, 0.71, 0.98]

Let us recall that the value of α controls the changes of the exponential function and this results in the scope of the interval. From formulas (3.6), (3.7) we can see that with the increasing value of α, Q will be decreasing. The exponential function tends to be 0 if α is assumed high values. If α is too small, the exponential function starts to become equal to 1 for its arguments. In practice, we set the value of α to be in the range of [0, 5].

Let us optimize the upper bound of the interval (the lower one is determined in an analogous manner) using the synthetic data given at the beginning of this section. The plot of Q for selected values of α on upper bound of the given example, see Figure 9 (a), shows that there is an optimal value of b associated with a clearly visible maximal value of Q for α = 0.1 and 0.01. As it could have been expected, higher values of α produce more specific information granules (shorter intervals). When the numeric values are associated with their individual memberships, the relationship between value of Q and upper bound is shown in Figure 9 (b). It exhibits a similar tendency but with a different location of "b" resulting from the optimization of Q.

When α is equal to zero, the values of denominator in (3.6) and (3.7) are equal to 1. So we only need to maximize numerator—to count the number of points in the entire range. If the bounds formed in this way cover all testing data, or most data (for example cover p% data, where p tends to be 100), we can say that the granular model comes with about round p% data being covered.



(a) (b)

Figure 9. Q versus b for different values of α for the numeric data: (a) numeric values without membership degrees (b) numeric values with membership degrees

3.3. Numerical Evaluation of Information Granules

We have discussed the principle of justifiable granularity in chapter 3.2. Here we consider a situation of satisfying two conflicting requirements—specificity and generality. To come up with a single performance index Q taken as a product of two functions (requirements) f_1 and f_2, Q $=f_1*f_2$ where f_1 is an increasing function of the cardinality of the elements falling within the bounds of the information granule Ω (generality) and f_2 is a decreasing function of the support (length) of |b-a| (specificity).

One of the interesting alternatives is to consider the following two functions:

$$f_1(u)=u \tag{3.8}$$

and

$$f_2(u)= e^{-\alpha u} \tag{3.9}$$

where u =|b-m| and α is a positive parameter controlling the changes of the function and subsequently offering some flexibility in the produced information granule. In the optimization problem, it is helpful to consider a two-step process. First, a numeric representative of the data is determined (say, a mean or median or any other numeric representative) and the formation of information granules is realized separately for the lower bound (taking all data that are lower than this numeric representative) and the upper bound (where we consider the data positioned to the right from the numeric representative). The maximization of the expression

$$V(a) = \text{card}(d|\Omega(d))*\ e^{-\alpha|m-a|} \tag{3.10}$$

is carried out with respect to unknown lower bound "a". Here "m" stands for the numeric representative of the numeric data. Likewise, we proceed with the maximization of the expression (objective function) V(b) = card(d|Ω(d))* $e^{-\alpha|m-b|}$

with “b” being the upper bound of the information granule. It is worth noting that (3.10) comes with some parameter (α), which is essential to its further usage in granular constructs.

With regard to the above maximization (3.10), one can note an interesting and intuitively appealing relationship between “b” and the values of the parameter α. Consider that the data larger that the numeric representative are d_1, d_2, …, d_N with “m” being a certain numeric representative. In spite of the values of α, the location of optimal “b” is in-between minimal d_i and maximal d_i. Given the form of (3.13), if b falls below the bound minimal d_i, then the performance index is equal to zero. If b moves beyond maximal d_i, the performance index decreases monotonically. Indeed, take b_1 and b_2 where $b_1<b_2$. In this case, the performance index $V(b_1)$ is higher that $V(b_2)$ which indicates that moving towards higher values of “b” after exceeding maximal d_i leads only to the reduced values of the performance index.

In a similar way we can develop an information granule in the form of a certain fuzzy set. Here the only difference is that the cardinality of the set of data contained within the interval has to be replaced by the sum of membership degrees of the data belonging to the fuzzy set.

To proceed with a numeric quantification of these two criteria, we introduce two indicators that are closely reflective of the two criteria we identified above. These two indicators are often used in our granular models in the following chapters.

(a) The cumulative length of information granules (intervals) of outputs of the granular model described as

$$L = \sum_{i} L_i \tag{3.14}$$

which is reflective of the specificity (the level of detail) achieved in the granular model, and L_i is the length of an interval generated by the i-th instance.

(b) Level of coverage of the data expressed as a ratio

$$I = \frac{\text{no. of data covered}}{\text{all data}} \tag{3.15}$$

The higher the ratio is, the better the coverage of the data. Ideally, we may like to see “I” being very close to 1 however this might result in an unacceptable lack of specificity of the results provided by the granular model. Accepting a certain value of α in the construction of the information granules through running the principle of justifiable granularity, we calculate the resulting values of I as well as the cumulative length of intervals of the granular outputs of the model.

An example plot in the I-L coordinates is shown in Figure 10. An important characterization of the quality of the granular model is obtained by computing an area under curve (AUC), that is

$$\int I(\alpha)d\alpha \tag{3.16}$$

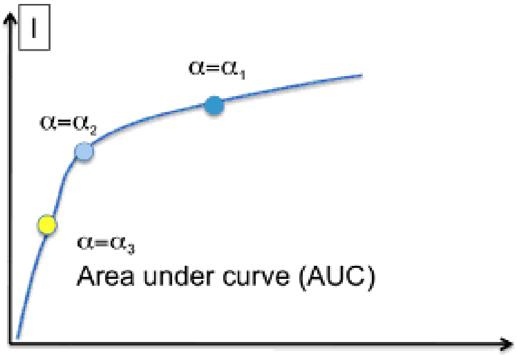


Figure 10. Plot of I versus accumulated length L for selected values of α, $\alpha_1 > \alpha_2 > \alpha_3$

The higher the value of the AUC, the better the obtained model is. The monotonicity property holds: higher values of α imply broader intervals and as a consequence, result in a higher coverage of the data. Note, however that the specificity of the results (intervals) is negatively impacted. A single numeric characterization of the curve I(α) is provided in the form of an AUC. It could be sought as a synthetic indicator of the performance of the granular model. In this way, two granular models can be compared: the one with the higher value of the AUC is preferred.

3.4. Granulation of Linguistic Terms

Given a numeric scale, 1 to 9 as an example, some linguistic descriptions are always pursuit in decision making problems. Linguistic terms used in human-centric systems like *low*, *medium* and *high* are a simple partition on this scale. They can be effectively represented by intervals and fuzzy sets. Schematically, we can portray the process of arriving at the operational representation of linguistic terms as illustrated in Figure 11. In this figure, capital letters denote the corresponding linguistic terms: L- *low*, M – *medium*, H – *high*.

The two important features of such granulation mechanisms are worth noting here: (a) the mapping is by no means linear that is a localization of the associated information granules on the scale; (b) the semantics of the terms allocated in the process of granulation are retained. Various information granulation formalisms can be contemplated including sets (intervals), rough sets, fuzzy sets, and shadowed sets (Pedrycz, From fuzzy sets to shadowed sets: interpretation and computing 2009), just to mention several alternatives.

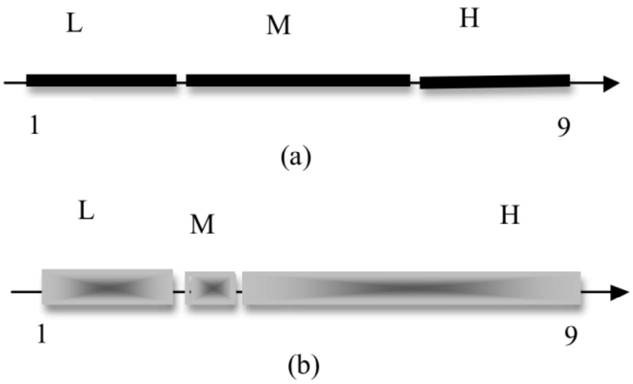


Figure 11. Towards the operational realization of linguistic terms: (a) realization with the aid of intervals, and (b) fuzzy set-based implementation

The question on how to arrive at the operational version of the information granules can be reformulated as a certain optimization problem. To achieve high flexibility when stating the optimization problem, especially expressing a minimized optimization criterion, we use a technique of Particle Swarm Optimization (PSO) (Kennedy and Eberhart, Particle swarm optimization 1995) (Ling, et al. 2008) (Trelea 2003) (Tripathi, Bandyopadhyay and Pal 2007) (Bergh and Engelbrecht 2006) (Wang and Watada 2010) as a viable vehicle. This population-based method offers a significant level of diversity of possible objective functions, which plays a role of fitness functions.

3.5. Conclusions

The algorithm realizing the principle of justifiable granularity produces an information granule (either an interval or a fuzzy set or other types) based on a collection of numeric data. The nature of the numeric data themselves can be quite different. Two situations are worth highlighting here:

(a) The numeric data could result from measuring some variables present in the system. In this case, information granules are treated as non-numeric data, which can be then used in the design of the model and highlight the structure of a large number of numeric data.

(b) The numeric data are just membership values of some fuzzy sets reported for a certain point of the universe of discourse. The granular representation resulting from the discussed construct gives rise to the information granule of higher type, fuzzy set of type-2, to be more specific. Depending on the nature of the information granule formed here, we construct an interval-valued type-2 fuzzy sets or type-2 fuzzy sets. It is worth stressing that in this manner, we have arrived at a constructive way of designing of type-2 fuzzy sets – the area that has been very much left neglected in the existing studies.

The algorithms supporting the design of information granules presented so far have been concerned with one-dimensional data. In situation of multivariable data, the developed method can be applied to individual variables separately.

4. Allocation of Information Granularity

The principle of justifiable granularity we introduced in previous chapter elaborates a way how to construct a meaningful representation of a collection of numeric values (real numbers)—information granules. It is based on real numbers that all are under the same definition, such as outputs of local models. When we are faced with a single model or a set of individual models, a proper way to provide elasticity or generalize a model is to consider allocating/distributing information granularity to their components. We will introduce the fundamental algorithm of the allocation of granularity, several frequently used protocols and two real application examples in this chapter.

4.1. Information Granularity and Its Allocation

For a model which has numeric parameters and structure, system modeling is realized in a more or less local way: the model is suitable for the data provided so far. Owing to the huge size of data stored in the database over time, an abstract (general) system model becomes an overwhelming alternative. An appealing way to solve this problem is to assign a level of granularity to the entire system. The concept of granularity is closely related with the "size", "capacity" or "dimension" of the information granule (Pedrycz and Gomide, Fuzzy Systems Engineering: toward human-centric computing 2007). From the intuitive point of view, the size of the granule describes its specificity. We say that how specific the granule is and how many details are embraced. The more elements we identify as belonging to the granule, the more abstract and general it becomes. The notion of cardinality is the one commonly used to address the problem at the corresponding level of generality (specificity). The granularity brings a flexibility to view the system. Moreover, the level of granularity provides an opportunity to avoid the noise and disturbance on components (parameters and structure) of the system caused by training data and learning methods. However, an inevitable issue appears simultaneously: how to distribute (allocate) the assigned level of granularity to components of the system?

This chapter serves as an exploration to a proper mechanism for allocation of granularity.

The level of information granularity depends heavily on the task at hand and the need of the decision-making process. Assuming that given a level of information granularity "ε" with its value in the unit interval [0, 1], it is going to be allocated to individual components of a system or local systems, local models, and alike. We name these components or local systems, local models as "units" and each unit is assigned with a numeric value of granularity which reflects the level of flexibility of the unit. For example, the allocation of granularity to a neural network is the task of allocating information granularity to its connections (weights and biases). Thus the value of the unit refers to a weight or bias. Since there may be a number of units, the process of allocation needs help from a certain optimization strategy, like evolutionary methods, PSO, ant colony, etc.

The essence of allocation of information granularity is to find a best combination of granularities that satisfies an optimization of an objective function which reflects the performance of the whole granular system. We require that the overall balance of granularity is retained, meaning that the requirement

$$h\varepsilon = \sum_{i=1}^{h} \varepsilon_i \tag{4.1}$$

is satisfied. The values of ε_1, ε_2, …, ε_h are levels of granularities for different units and become the core of the overall optimization problem. h is the total number of units. Since the values of ε_1, ε_2, …, ε_h may differ a lot, different units show different levels of granularity--generality and specificity. However, we need an overall criterion of the whole system or model to compare with others. That is why we ask for the fulfilment of (4.1). In this way, the value of ε is seemed as information granularity of the whole system or model. We will use symbol ε for the same meaning in the following chapters. With regard to different formations of granules, the protocols of allocation of granularity have some diversity.

4.2. Fundamental Protocols of Allocation

The issue of the allocation of information granularity to individual units can be realized in several different ways depending on how much diversity one would like to accommodate in the allocation process.

Let's first look at protocols of allocation when information granules are represented by intervals which are basic and helpful for the more complicated types of information granules, refer to Figure 12. The following five fundamental protocols are studied; note that in all cases the balance of granularity presented above is satisfied (4.1).

C_1: uniform allocation of information granularity. This protocol is the simplest one. Each unit is affected in the same way. In essence, this allocation does not call for any optimization. All units become replaced by intervals constructed with the use of the same value of ε. Intervals themselves are distributed symmetrically around the original numeric value of the unit meaning that we have $\varepsilon_- = \varepsilon_+ = \varepsilon/2$. Note that ε_- is the granularity for the lower part of the interval and ε_+ is for the upper part.

C_2: uniform allocation of information granularity with asymmetric position of intervals around the original units. In this case, each unit uses the same level of granularity that is ε. Instead of $\varepsilon_- = \varepsilon_+ = \varepsilon/2$ for all units, ε_- and ε_+ may be different. For each unit, the condition of $\varepsilon_- + \varepsilon_+ = \varepsilon$ is also satisfied. That is: $\varepsilon_- = \chi*\varepsilon$, and $\varepsilon_+ = (1-\chi)*\varepsilon$. χ is a value between 0 and 1. Same rule is abided by the following protocols.

C_3: non-uniform allocation of information granularity with symmetrically distributed intervals of information granules. The levels of granularity are different for different units and equal to ε_i. Because of the symmetry, we have $\varepsilon_{i-} = \varepsilon_{i+} = \varepsilon_i/2$. Note that ε_{i-} is for the lower part of the ith interval and ε_{i+} is for the upper part.

C_4: non-uniform allocation of information granularity with asymmetrically distributed intervals of information granules. The levels of granularity of different units may be different and we have $\varepsilon_{i-} + \varepsilon_{i+} = \varepsilon_i$, where $\varepsilon_{i-} = \chi_i*\varepsilon$, and $\varepsilon_{i+} = (1-\chi_i)*\varepsilon$.

C_5: random allocation of information granularity. This protocol is proposed as a reference method to demonstrate how much improvement is achieved by optimizing the granular units by using different protocols.

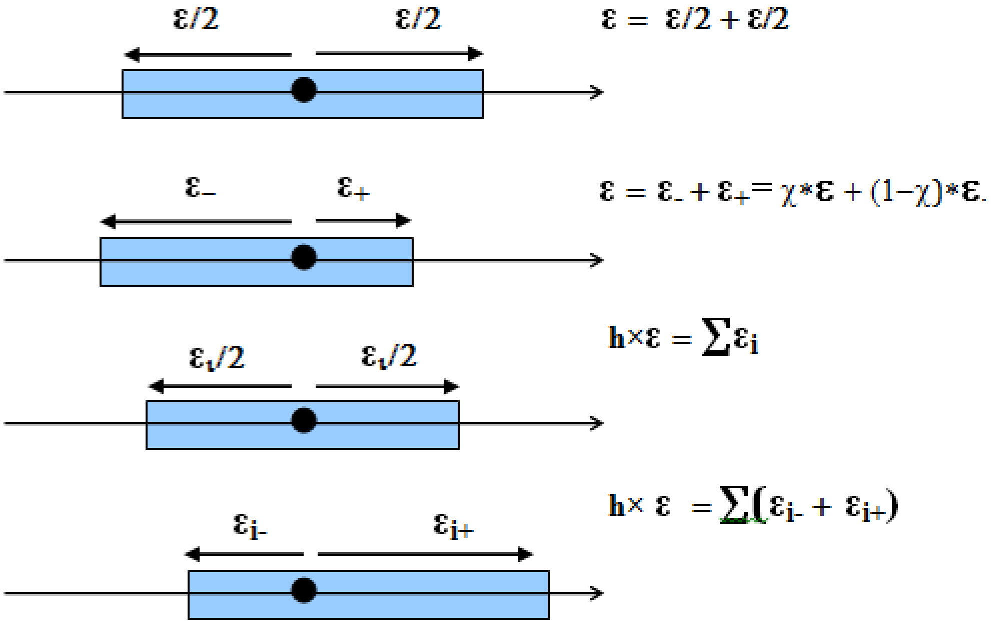

$\varepsilon/2$
$\varepsilon/2$
$\varepsilon = \varepsilon/2 + \varepsilon/2$
ε_-
ε_+
$\varepsilon = \varepsilon_- + \varepsilon_+ = \chi*\varepsilon + (1-\chi)*\varepsilon.$
$\varepsilon_i/2$
$\varepsilon_i/2$
$h\times\varepsilon = \sum\varepsilon_i$
ε_{i-}
ε_{i+}
$h\times \varepsilon = \sum(\varepsilon_{i-} + \varepsilon_{i+})$

Figure 12. Protocols for intervals and their associated constraints

Each protocol presented above implies a certain way to realize allocation of information granularity. In the case of C_1 and C_2, we envision no optimization or a very limited one-dimensional optimization. As to protocols C_3 and C_4, the allocation process has to be realized through some optimization techniques since the problem is of high dimensionality. Given the increasing level of flexibility in the realization of the granularity allocation protocols, C_4 offers the highest level of flexibility.

By these protocols, the resulting interval granule is described as ***P*** (u) = $[u^-, u^+]$.

$$u^- = u - \varepsilon_- \tag{4.2}$$

$$u^+ = u + \varepsilon_+ \tag{4.3}$$

This has a premise that the original numeric values of units are mapped into [0, 1].

4.3. Allocation of Granularity in Neural Networks

Neural networks as a fundamental architecture in system modeling is used here as a typical example to show how to realize allocation of granularity. Although we are confined with a specific topology—a multilayer perceptron (MLP) (Rocha, Cortez and Neves 2005), this strategy applies equally well to all other topologies, such as radial basis function neural networks (RBFNN), neurofuzzy systems and so on.

Assume that a numeric network that has already been developed by means of one of well-established learning strategies. Then a data set is used to construct a granular network, viz. form interval connections on a basis of the given network. The essential part during this process is the allocation of information granularity.

The granular neural network under consideration, presented in Figure 13, comprises a single hidden layer consisting of n_1 neurons, and a single neuron located in the output layer. Features (inputs) to the network are organized in a vector form $\mathbf{x} = [x_1, x_2, ..., x_n]^T$. The weight (connection) connecting the ith neuron in the input layer to jth neuron in the hidden layer comes in the form of an interval and is denoted by W_{ji}, $W_{ji} = [w_{ji}^-, w_{ji}^+]$. The weights between hidden layer and output layer are also interval-valued. Each neuron in the hidden layer and output layer comes with an interval bias. In virtue of the interval connections used in the network, for any numeric input, the result of processing is an interval, say $Y = [y^-, y^+]$.

The challenging yet highly important issue is how to construct interval-valued weights and biases of a network. The available information granularity (more specifically, its level of granularity), being treated as an important design asset, has to be carefully distributed among all the connections of the network so that the interval-valued output of the neural network "covers" (includes) the experimental datum. In what follows, we use the proposed protocols of allocation of information granularity and discuss two indexes whose optimization is realized through this allocation process.

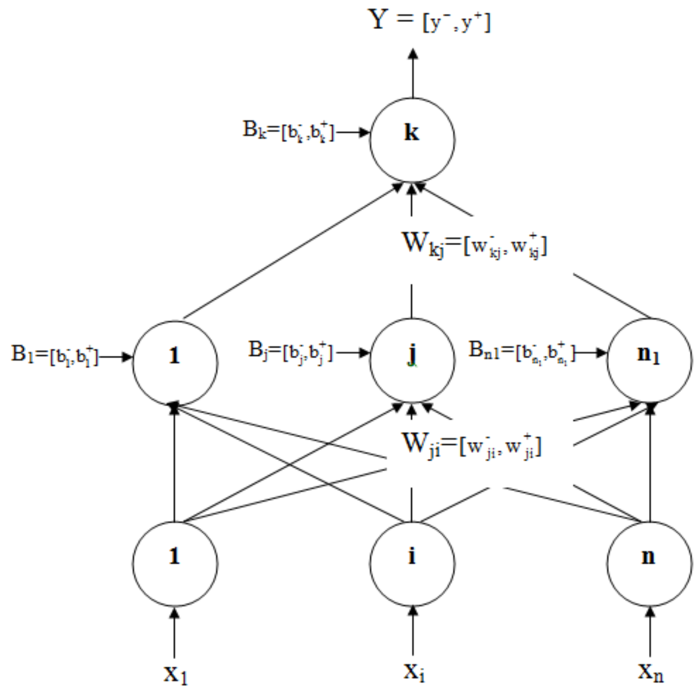


Figure 13. An architecture of a granular neural network

Given a level of information granularity ε assuming values in the unit interval, it is allocated to the individual weights and biases. A way of building intervals around the original numeric values of the weights and biases can be referred to as granularity allocation. The allocation leading to the optimization of a given performance index refers to *optimal* information granularity allocation. The original weight or bias, denoted symbolically by “w_{ji}” and "b_k" is made of interval character by forming some bounds around the original numeric values in the following way

$$w_{ji}^{-} = w_{ji} - \varepsilon_{-} |w_{ji}| \text{ and } b_k^{-} = b_k - \varepsilon_{-} |b_k| \qquad (4.4)$$

$$w_{ji}^{+} = w_{ji} + \varepsilon_{+} |w_{ji}| \text{ and } b_k^{+} = b_k + \varepsilon_{+} |b_k| \qquad (4.5)$$

The resulting interval granule of the weights and biases are described as ***G*** (w_{ji}) =W_{ji}= [w_{ji}^{-}, w_{ji}^{+}], or ***G*** (b_k) = [b_k^{-}, b_k^{+}]. In this notation, we use ***G*** to stress an operation of the formation of the granular connection. For a given value of ε and a total number of weights and biases equal to “h”, the overall balance of granularity is equal to hε, namely

$$h\varepsilon = \sum_{i=1}^{h}(\varepsilon_{i-} + \varepsilon_{i+}) \qquad (4.6)$$

This formula is an extension of (4.1).

The allocation of information granularity to individual connections can be realized in several different ways depending on how much diversity one would like to accommodate in the allocation process. Each protocol presented in chapter

4.2 (C_1 to C_5) implies a certain way to realize allocation of information granularity. In the case of C_1 and C_2, we envision no optimization or a very limited one-dimensional optimization. As to protocols C_3 and C_4, the allocation process has to be realized through some optimization techniques since the problem is of high dimensionality. Given the increasing level of flexibility in the realization of the granularity allocation protocols, C_4 offers the highest level of flexibility.

4.4. Allocation of Granularity in Analytic Hierarchy Process

Another example of allocation of granularity being a flexible asset in system modeling is in group decision making problems in which each decision maker (reciprocal matrix) is assigned a level of granularity (we denote it as α here). With flexibility, each decision maker can adjust their pairwise results and choose the best representative reciprocal matrix that reaches high consensus among all decision makers.

Consider a group decision-making scenario in which there are "c" decision-makers and each of them comes with own preferences (preference vectors), **e**[1], **e**[2],…, **e**[c] obtained by running the AHP for the corresponding reciprocal matrix R[1], R[2], …, R[c]. Each reciprocal matrix has the following expression: $R = [r_{ij}]$ and is provided with a finite scale with values spread in between 1 and 9. Furthermore the quality of preference vectors is quantified by the associated inconsistency index $\nu[i]$.

The very much needed elasticity/flexibility is brought into the AHP structures by allowing the reciprocal matrices to be *granular* rather than numeric. By granular reciprocal matrices we mean matrices whose entries are not plain numbers but information granules, say intervals, fuzzy sets, rough sets, probability density functions and others. In a nutshell, information granularity present here serves as an important modeling asset, which brings forward an ability of the decision-maker to exercise some flexibility to be used in modifying his/her own position when becoming aware of the preferences of the other members of the group. In essence, the reciprocal matrix is elevated (abstracted) to its granular format. To emphasize this, we use the notation ***G***(R) to underline the fact that we are concerned with granular reciprocal matrices where ***G***(.) stands for a specific granular formalism being used here. Being more specific, we can talk about interval-valued reciprocal matrices where ***G***(R) = ***P***(R). If information granularity is formalized in terms of fuzzy sets, then we have ***G***(R) = ***F***(R). When considering shadowed sets ***S***(.) (Korpela and Tuominen 1996) (van Laarhoven and Pedrycz 1983), we have ***G***(R) = ***S***(R).

Evidently, the higher level of granularity is provided to the individual decision-maker, the higher the likelihood of arriving at preferences accepted by all members of the group. Here we appealed to the intuitive concept of granularity by trying to present a qualitative nature of the process in which the asset of granularity is involved. This idea can be formalized depending on the form of information granules being the entries of the reciprocal matrices. For instance, if the granularity of information is articulated through intervals, the length of such intervals–entries of the reciprocal matrix can be sought as a level of granularity. In case of fuzzy sets, the level of granularity can be gauged by σ-counts of fuzzy sets (fuzzy numbers) that are the entries of the granular (fuzzy) reciprocal matrix. In our further considerations, we confine ourselves to interval-valued reciprocal matrices with the level of granularity equal to α. The flexibility offered by the level of granularity can be effectively used to optimize a certain objective function capturing the essence of the reconciliation of the individual preferences.

An idea of having entries of the reciprocal matrix R treated as granular entities was originally introduced in (van Laarhoven and Pedrycz 1983), see also (Wang and Chen 2008). More specifically, in this study discussed was a generalized version of the AHP where the reciprocal matrices were composed of triangular fuzzy numbers. The underlying motivation was to reflect uncertainty present in judgments of decision-makers when forming pairwise estimates. The one advocated here exhibits some resemblance in the sense we are again concerned with the idea of granularity of information. The crucial difference lies in the treatment of information granularity – here we treat it as a conceptual vehicle to facilitate admissible changes to the results of pairwise comparison. This fact is underlined by including a granular form of the reciprocal matrices allowed in the problem, that is R[1] R[2], …, R[c] which are elements of the family of interval-valued reciprocal matrices, namely ***P***(R).

Two suitable scenarios of allocation of granularity among decision-makers are considered here, namely a uniform allocation of granularity and non-uniform distribution of granularity. For uniform allocation, starting with the initial reciprocal matrix provided by the expert and assuming a given level of granularity α (located in the [0, 1] interval), let us consider an entry for which r_{ij} is lower than 1. The interval of admissible values of this entry of ***P***(R) implied by the level of granularity is equal to

$$[a, b] = [\max(1/9, r_{ij} - \alpha\,(8/9)), \min(1, r_{ij} + \alpha\,(8/9))] \qquad (4.7)$$

As to non-uniform allocation, each reciprocal matrix is assigned a level of granularity α_i. ***P***(R) is equal to

$$[a, b] = [\max(1/9, r_{ij} - \alpha_i\,(8/9)), \min(1, r_{ij} + \alpha_i\,(8/9))] \qquad (4.8)$$

It is interesting to observe how the allowed level of granularity α translates to the maximal changes in the integer entries of the reciprocal matrices. Table 1 offers a view at this relationship – this is also helpful in envisioning a reasonable range of granularity which exhibits a sound impact on the modifications of the entries of R's.

Table 1. Maximal changes to the entries of reciprocal matrix versus assumed values of the granularity level

α	0.005	0.01	0.015	0.02	0.025	0.03	0.05
maximal changes to entries of R	0	1	1	1	2	2	3

4.5. Conclusions

This chapter illustrates several basic protocols of allocation of granularity and explains their potential application fields. One is in the abstraction of an existing architecture like neural networks (no matter what types) through optimal allocation of granularity to their connections guided by an optimization criterion. With the granular architecture of a system one can effectively gauge the performance of the already constructed model in presence of training data and when dealing with new data, it is instrumental in the quantification of the quality of the obtained result. Another application case in system modeling is concerned with dealing with group decision making problems where we provide each decision maker some flexibility (a level of granularity). The usage of information granularity reflects upon and quantifies the diversity among decision makers within a group. Those two real world application examples will be detailed discussed in the subsequent chapters with aid of experiments.

5. A Genetic Reduction of Feature Space in the Design of Local Fuzzy Models

Before we proceed to system modeling with granular architecture, let’s look at a strategy frequently used to deal with raw data which are sent to modeling: reduction of feature space. It becomes more important in various modeling methods as a transparent yet accurate model is often required by users and reduction strategies can help improve transparency and accuracy. Feature reduction is a straightforward yet effective method that we will discuss its main advantages in the follows. We realize fuzzy modeling with the reduced feature space and finally a functional rule-based fuzzy model is constructed.

5.1. Reduction of Input Space

The challenge of building transparent yet accurate enough models remains in the area of fuzzy modeling. This is particularly essential considering that retention or enhancements of interpretability is often difficult to realize when the dimensionality of the input space increases. Fuzzy models, in one way or another, rely on the use of information granules – fuzzy sets in the formation of their structure (Alcala, et al. 2009) (Delgado, Gomez-Skarmeta and Martin 1997) (Park, Pedrycz and Oh 2009). Information granules constitute a backbone of these models based on which further details are developed. For instance, in rule-based models a great deal of attention is given to fuzzy sets forming the condition parts of the consecutive rules. The functional (linear) models forming the conclusion parts and a way of rule aggregation are the subject of further detailed design steps. Information granules link also with the important aspect of interpretability of fuzzy models relating in an important way with the easiness that the model can be perceived by the user and the underlying nature of the modeled phenomenon described in a clear way. The effect of combinatorial explosion of features comes into the picture.

There is no surprise that there have been a variety of ways in which the number of input variables (features) can be reduced. This happens either by running feature reduction or feature transformation mechanisms. The reduction of the number of input variables is one of the appealing alternatives to enhance the compactness of the model. Then a construction of information granules is realized by a clustering mechanism in the reduced input space. Hence the term “information granularity” is referred to the number of clusters. Evolutionary optimization, including Genetic Algorithms, Particle Swarm Optimization and Ant Colonies, has emerged as one of interesting alternatives supporting the design of fuzzy models (Bonissone, et al. 2006) (Pham and Castellani 2006).

By emphasizing the essence of the concept of information granules and information granulation permeating the area of system modeling, the main objective of this study is to come up with a coherent design framework in which reduction of input space, granulation of information and realization of functional models result in cohesive and intertwined activities. By exploiting the well-known mechanisms, it is our intent to show that when being used in a coherent manner along with carefully utilized capabilities of parametric optimization they support the design of accurate and interpretable fuzzy models. As it is well-known in the literature, feature reduction (reduction of input space) is one of the essential mechanisms at arriving at fuzzy models of elevated interpretability as well as higher approximation abilities.

The objective of this study is to inherently associate the process of reducing the feature space with the design of fuzzy models by invoking the genetic optimization in the wrapper mode of feature selection. With this regard, there are several design issues that deserve attention and have to be carefully quantified with some potential design guidelines:

(a) Relationships between granularity of information processed in fuzzy models and associated abilities of the input space reduction,
(b) Impact and effective use of parameters of fuzzy clustering in the realization of the fuzzy model in the reduced input space,
(c) A level of interaction between the rules forming the fuzzy model and the limits of dimensionality reduction

To fully concentrate on the reduction process itself, we intentionally embark on the well-established topology of fuzzy rule-based models as those are thoroughly documented in the literature and as such can be used as a convenient experimentation platform. Dimensionality reduction is usually realized in two different ways: (a) by reducing the feature (input) space, and (b) by transforming the original feature space. In this study, we confine ourselves to the reduction of the feature space as it retains the identity of the individual input variables. In contrast, feature space transformation could be more effective (as it does not invoke any combinatorial optimization, which is evidently associated with a selection of a subset of features encountered in any reduction of the input space). It, however, makes the features built in the transformed space difficult to interpret.

5.2. Key Design Phases of Fuzzy Models

One of the essential design phases of the model is the dimensionality reduction of the input space, which becomes critical in all cases where there are a large number of inputs and we are concerned with the enhanced interpretability of fuzzy models. To alleviate the problem, the original input space becomes reduced by retaining only a subset of inputs that are deemed to be the most essential from the modeling perspective. In the sequel a set of information granules $\{A_i\}$ is formed in the reduced space and the associated local linear models become constructed. We envision a general flow of design as portrayed in Figure 14. In contrast to the standard development of fuzzy models being encountered in the literature, the essential phase present here deals with dimensionality reduction of input (feature) space. The fuzzy model described here operates in the reduced space.

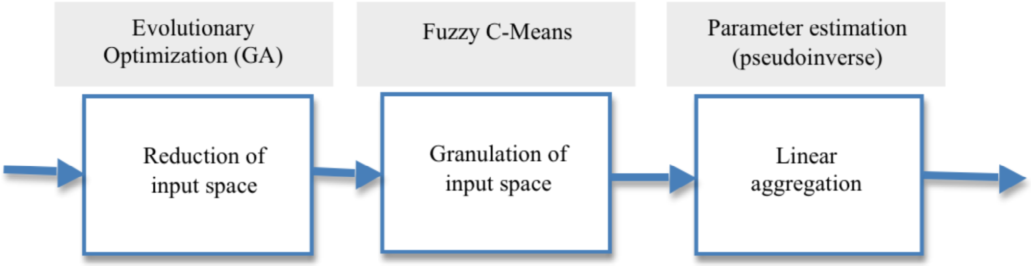

Evolutionary Optimization (GA)
Fuzzy C-Means
Parameter estimation (pseudoinverse)
Reduction of input space
Granulation of input space
Linear aggregation

Figure 14. An overall flow of design of the fuzzy model and its association with its underlying architecture and ensuing mechanisms of unsupervised and supervised learning

The design process naturally implies the use of suitable optimization tools. The role of various techniques arising under the rubric of Evolutionary Optimization and swarm-like optimization techniques are worth stressing, see e.g. (Araujo and Coelho 2008) (Zhao, et al. 2010) (Oh, Pedrycz and Roh 2009).

Here, in contrast to the commonly existing design practices, we engage a broad spectrum of techniques of structural as well as parametric optimization. In what follows, we highlight the nature of the optimization processes. Different learning phases are realized in unsupervised and supervised modes. The experimental data are provided in the form of input-output pairs ($\mathbf{x}'_k$, y_k), k = 1, 2, …, N, where $\mathbf{x}'_k \in \mathbf{R}^n$ and $y \in$ R. The reduction of the input space results in the pairs ($\mathbf{x}_k$, y_k) where now $\mathbf{x}_k \in \mathbf{R}^p$ and $p < n$. When describing each of the design phases, we stress a suite of parameters, which can be used to assure a sufficient level of the design flexibility of the overall design.

5.2.1. The architecture of the functional models

The architecture of our fuzzy model is dwelled upon a collection of information granules, which are formed in the input space. These information granules – fuzzy sets form a nonlinear transformation of the input space into the c-dimensional unit hypercube where any input **x** results in a vector of activation levels – membership grades of information granules A_1, A_2, …, A_c. In more detail, we use the notation $u_i(\mathbf{x}) = A_i(\mathbf{x})$ to emphasize that the activation level produced by A_i has been implied by **x**. The choice of these fuzzy sets forms a suitable cognitive perspective at which the model is being formed. With each information granule we associate a functional (linear) model $L_i(\mathbf{x}, \mathbf{a}_i) = \mathbf{a}_i^T\mathbf{x}$ with $\mathbf{a}_i$ being the vector of the parameters of the i-th linear model. The aggregation of these models to form an overall input –output relationship is governed by the well-known relationship

$$y(\mathbf{x}) = \sum_{i=1}^{c} A_i(\mathbf{x}) L_i(\mathbf{x}, \mathbf{a}_i) \tag{5.1}$$

where y(**x**) is the output of the fuzzy model for **x**. The derivation process of the vector of parameters is elaborated in chapter 2.2. From the point of view of flow of processing, there are two phases: (a) nonlinear transformation of input space into a vector of degrees of membership in the c-dimensional unit hypercube, and (b) linear aggregation of the membership degrees.

5.2.2. Development of reduced feature space

In order to deal with feature searching space's explosion problem when the number of features is large, some feature dimensionality reduction approaches have been proposed. For many practical problems, the original data sets are obtained without professional analysis and this makes the noisy features' existence. Our experiments show that only subset features may have the same or similar performance (accuracy) with using all the features on regression problems.

In this section, we do the feature selection with evolutionary algorithms (EA) and genetic algorithms (GAs), in particular. EA has already been used for structural and parameter optimization in fuzzy modeling. Real-code GA is adopted as the technique to select essential features. In order to decide the number of subset features, we have done experiments from one feature to all features on some benchmark data sets. All the results show the same tendency that there exists a critical drop on output error and after which the error just fluctuates in a very small range. In this way, we can say that this point or its next point (or we can give a criterion to decide the point) is the number we would like. At the same time we also know which features are selected. At this point, the prep-processing has been completed. The subset features can be used as the input in the next step in fuzzy modeling.

Selecting a relevant subset of features (inputs) from a large set of features is critical to the effective construction of fuzzy models, their accuracy and interpretability. The problem is of combinatorial character. Selection "p" out of "n" requires the number of attempts equal to

$$\binom{n}{p} = \frac{n!}{p!(n\text{-}p)!} \tag{5.2}$$

which for large values of "p" calls for some optimization. Genetic Algorithms form one of the viable alternatives. As they are well documented in the literature, we recall only the most essential realization aspects. The real-number content of the chromosomes is more suitable to represent the search space of subsets of features. The length of the chromosome is equal to "n". The decoding of the chromosome is realized as follows. The entries are ranked and replaced by integer values of the ranking indexes. The string of integers obtained in this way produces the subset of indexes of the input variables which form the reduced space. The fitness function describes the performance of the fuzzy model; the relevant details will be covered in the subsequent sections.

5.2.3. Design of information granules

A commonly accepted way of forming information granules (Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic 1997) on a basis of experimental data is to use techniques of fuzzy clustering. With this respect, Fuzzy C-Means (FCM) (Bezdek 1981) has assumed a highly visible position. As the method is well documented in the literature and has been exploited quite often in the context of fuzzy modeling, rather than presenting details, we recall the most essential design aspects. The minimization of the objective function of (2.7) is realized in an iterative fashion by adjusting location of prototypes and the partition matrix. The two important design parameters here concern the number of fuzzy sets and a value of the fuzzification coefficient (m) assuming values larger than 1. The distance function is predefined in advance and the iterative scheme given in the literature works under the assumption of the use of the Euclidean distance or its weighted counterpart.

The fuzzification coefficient implies a certain level of interaction between the rules. The higher the overlap between the information granules, the more interaction occurs between the granules. This increased interaction is manifested at the level of the rules themselves, which has a detrimental effect on the interpretability of the rules, which cannot be easily interpreted out of the overall context formed by the collection of the rules. The interaction between two rules (their conditions) A_i and A_j can be quantified in the form of the integral

$$f(m) = \int \min(A_i(x), A_j(x))dx \tag{5.3}$$

or, which becomes more practically viable, as a sum taken over all data used to construct the fuzzy model

$$f(m) = \sum_{k=1}^{N} \min(A_i(x_k), A_j(x_k)) \tag{5.4}$$

The relationship between the values of "m" and the interaction level expressed as a function of m, f(m), is illustrated in Figure 15.

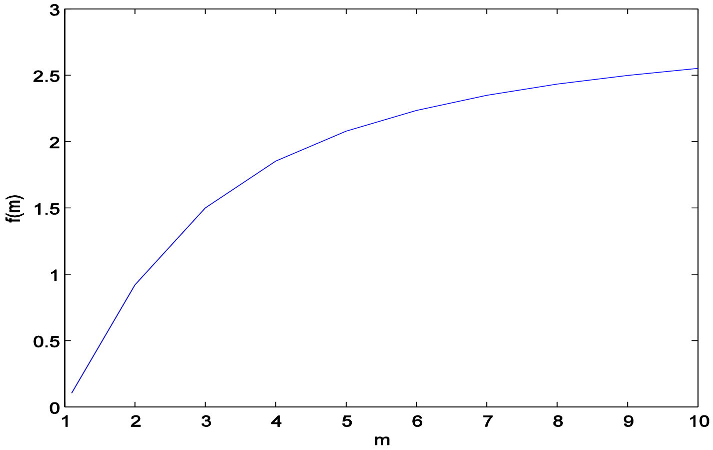


Figure 15. An example plot of the interaction index regarded as a function of the fuzzification coefficient (m). The prototypes are defined in one-dimensional space where v_1 = 1.0 and v_2 =7.0.

The interaction between the rules is indirectly associated with their interpretability; the higher the interaction level, the lower the interpretability of the rule in the sense that when displaying the rule, we have to be cognizant that its performance does not only depend on what it contains in the condition and conclusion parts but also relates to the contributions coming from other rules and being brought forward to the picture through the particular level of interaction. Given the relationship presented in Figure 15, high values of the fuzzification index are not suitable from the interpretation perspective.

The granular signature of the data can be concisely viewed as a collection of fuzzy sets $\{A_i(\mathbf{v}_i, m)\}$. An interesting generalization of fuzzy clustering arises when for each cluster we allow for a different value of the fuzzification coefficient (Pedrycz, Knowledge-Based Clustering: From Data to Information Granules 2005). In other words, the objective function reads as follows

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m_i} \left\| \mathbf{x}_k - \mathbf{v}_i \right\|^2 \tag{5.5}$$

i = 1, 2, …, c. where $m_1, m_2, \ldots, m_c > 1$. The optimization is realized in a slightly different way in comparison to what has been done so far. While one cannot determine the partition matrix in the same fashion as in the standard FCM, we postulate the calculations to be in the form

$$u_{ik}(t+1) = \frac{1}{\sum_{j=1}^{c} \frac{\left\| \mathbf{x}_k - \mathbf{v}_i(t) \right\|^{2/(m_i-1)}}{\left\| \mathbf{x}_k - \mathbf{v}_j(t) \right\|^{2/(m_j-1)}}} \tag{5.6}$$

while the prototypes are computed in the same way as before that is,

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m_i} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m_i}} \tag{5.7}$$

The associated advantage comes with the higher level of flexibility captured by vector of the fuzzification coefficients which as results produces the granular signature of the data to be in the form $\{A_i(\mathbf{v}_i, m_i)\}$. The drawback is more demanding optimization process to arrive at the optimal values of the entries of the vector $\mathbf{m} = [m_1 \ m_2 \ldots m_c]^T$.

5.3. Experimental Studies

Through the series of experiments reported in this section, we quantify the performance of the model and analyze the resulting structure as well as draw several observations pertaining to the overall design process and discuss an impact of various parameters of the model on its performance.

All experiments are carried out in the 10 fold cross-validation mode, which brings about higher stability of results in comparison with the case of a simple split of data in the training and testing part. The structural optimization of the model (selection of a subset of input variables) is realized through the use of the floating-point version of Genetic Algorithm (GA).

The values of the essential parameters of the GA such as the size of the population and the number of generations are selected experimentally based on some preliminary runs of the method. The content of each chromosome is translated into a subset of selected input variables to be used in the model. Given “n” dimensional inputs, the chromosome consists of “n” real number entries assuming the values in the [0, 1] interval. To select a subset of “p” inputs, we rank the entries of the chromosome, order them from the highest to the smallest and select those features whose indexes are the first “p” entries of the chromosome. Assume, for instance, that the content of the chromosome is [0.59 0.10 0.23 0.66 0.14 0.98 0.50]. The ranking leads to the order [5 1 3 6 2 7 4] and if p = 3 features are required; those are 5, 1, and 3. The design of the information granules is carried out in this reduced feature space. Once the information granules have been constructed and the associated local linear models developed, its performance is evaluated and it is being used as a fitness function. We use the elitist strategy, two-point crossover operation and a uniform mutation operation. The values of the crossover and mutation rates are equal to 0.8 and 0.05, respectively. They are in line of the values being encountered in the literature. Data sets used here are: synthetic data, auto MPG and Bodyfat.

Synthetic data

The two-dimensional synthetic data are generated from the nonlinear relationship of the form:

$$f(x_1, x_2) = 0.8*\sin(x_1) + 0.2*\sin(2*x_2) \qquad (5.8)$$

with the inputs assuming values in-between -4 and 4 (x_1), 10 and 18 (x_2). The plot of the data set generated by this relationship is shown in Figure 16. The training and testing data comprise 240 and 160 input-output data being uniformly distributed throughout the input space.

As we have only two input variables, there is no need to carry out genetic optimization to select a subset of input features. Considering both input variables, we run a series of experiments to determine an optimal value of the fuzzification coefficient “m” assuming a certain number of clusters. The results are presented in Figure 17. It has been found that the performance index depends on the values of the fuzzification coefficient; however, the general tendency of the relationship varies depending upon the level of granularity. For c = 2, there is a clearly visible minimum in the vicinity of c = 2. For c = 3 the relationship is quite monotonic with the values of Q being lower with the increase in the values of “m”; however the minimal value is obtained at m = 9.3. With further increase of the values of “m”, the values of Q increase; however there is some level of oscillations. For c = 4, there is a minimum m located around 5, Figure 17 (c). Higher values of the fuzzification coefficient imply a spike-like form of the membership functions, which assume values equal to 1 for the prototypes of the clusters and then rapidly move down. The average of membership degrees in each cluster tends to 1/c. The shapes of information granules do not change significantly when moving from c = 3 to c = 5, see Figure 17.

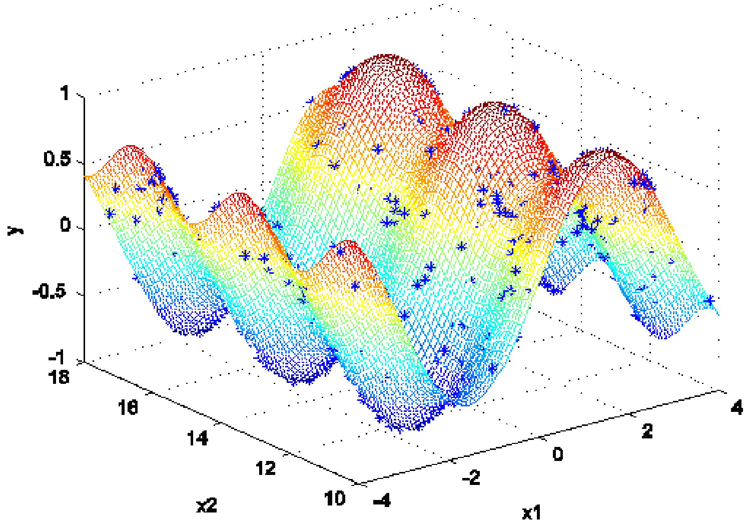
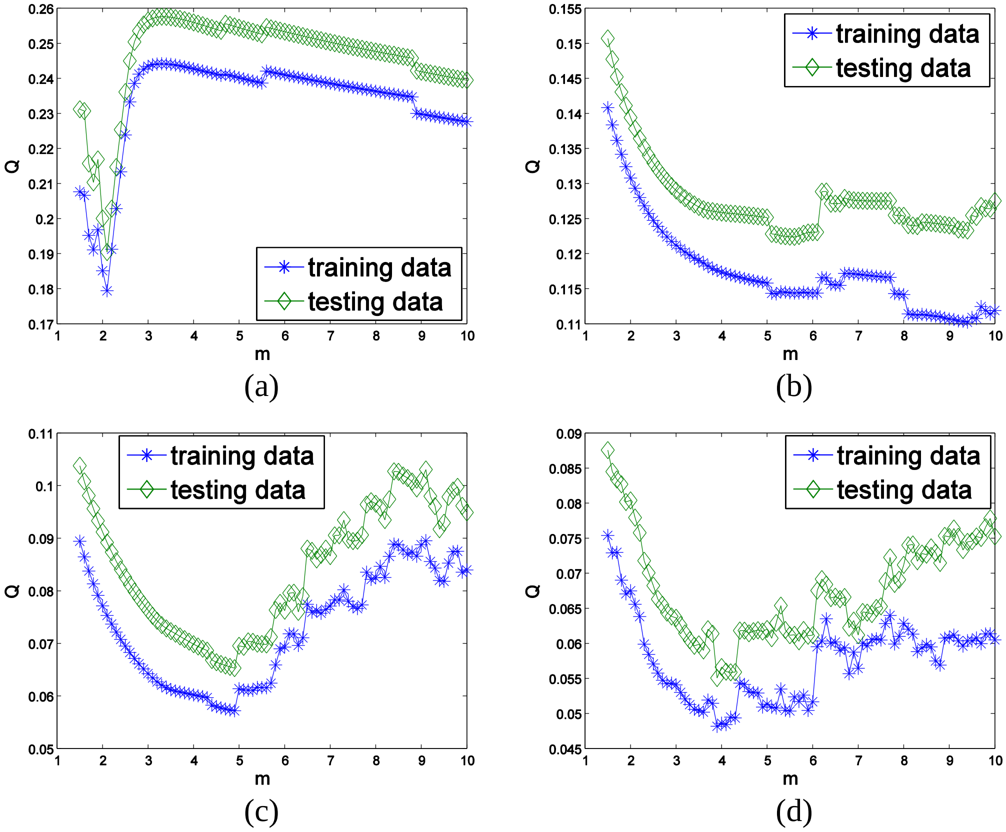


Figure 16. Two –dimensional input-output relationship used to generate synthetic data



(a) (b)

(c) (d)

Figure 17. Performance index Q as a function of "m" – results shown for the training and testing data, c = 2, 3, 4, 5

The distribution of the information granules for several selected values of "c" is illustrated in Figure 18.

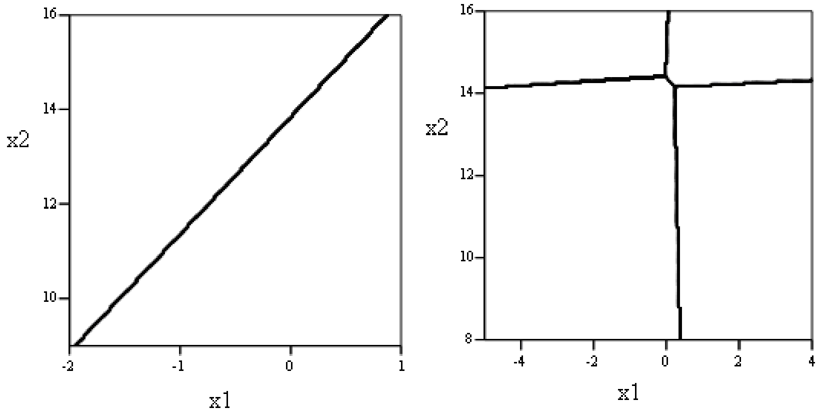


Figure 18. Distribution of information granules in the input space; shown are the boundaries of the clusters for c =2 and c =4.

The obtained results, which are reported vis-à-vis varying values of "c" are presented in Table 2. Here we include the values of the performance index for the training and testing set (both the mean values and standard deviations, std) and the optimal values of "m" (along with their means and standard deviations)

Table 2. Performance of the fuzzy model (training and testing data) for selected levels of granularity (number of clusters)

c	2	3	4	5
Q_{train} ±std Q_{test} ± std Optimal value of "m"	0.1142 ±0.0070 0.1294 ±0.0096 6 ±0	0.0663 ±0.0055 0.0731 ±0.0068 4.8500 ±1.7803	0.0222 ±0.0069 0.0268 ±0.0087 4.2000 ±1.7512	0.0163 ± 0.0037 0.0222 ± 0.0062 6.1100 ±2.4379

Now we reduce the feature space to a single variable; as shown in Table 3, it is apparent that the first variable was the preferred choice resulting in lower values of the performance index. The obtained results are reported in Table 3.

Table 3. Performance of the reduced fuzzy model (training and testing data; successive rows) for selected levels of granularity (number of clusters)

c features	2	3	4	5
First feature optimal value of 'm"	0.0621 ±0.0034 0.0652 ±0.0042 6 ±0	0.0273 ±0.0013 0.0290 ±0.0020 4.6500 ±1.8416	0.0233 ±0.0011 0.0265 ±0.0016 3.3500 ±1.8265	0.0196 ±0.0009 0.0226 ±0.0016 3.7000 ±1.3984
Second feature optimal value of 'm"	0.1923 ±0.0106 0.1987 ±0.0105 2.2500 ±0.6770	0.1821 ±0.0059 0.1876 ±0.0111 3.3000 ±1.7029	0.1676 ±0.0091 0.1788 ±0.0104 3.6000 ±1.5951	0.1696 ±0.0054 0.1840 ±0.0110 5 ±1.6330

When for each cluster we allow a different value of the fuzzification coefficient, and the optimized entries of the vector **m** = $[m_1\ m_2\ \ldots m_c]^T$ are obtained by real coded GA, we can find the same trend as above, just a slightly different of the "m" value. Please see Tables 4 and 5.

Table 4. Performance of the fuzzy model for the whole feature space (two features) and optimized values of the fuzzification coefficients of the individual clusters

c	2	3	4	5
Q_{train} ±std Q_{test} ± std range of values of "m"	0.1140 ±0.0071 0.1380 ±0.0117 [5.85 5.85]	0.0621 ±0.0056 0.0922 ±0.0184 [5.2 5.25 5.45]	0.0182 ±0.0060 0.0562 ±0.0241 [4.9 4.35 4.9 4.55]	0.0158 ± 0.0040 0.0481 ± 0.0249 [4.15 4.3 4.5 4.3 4.4]

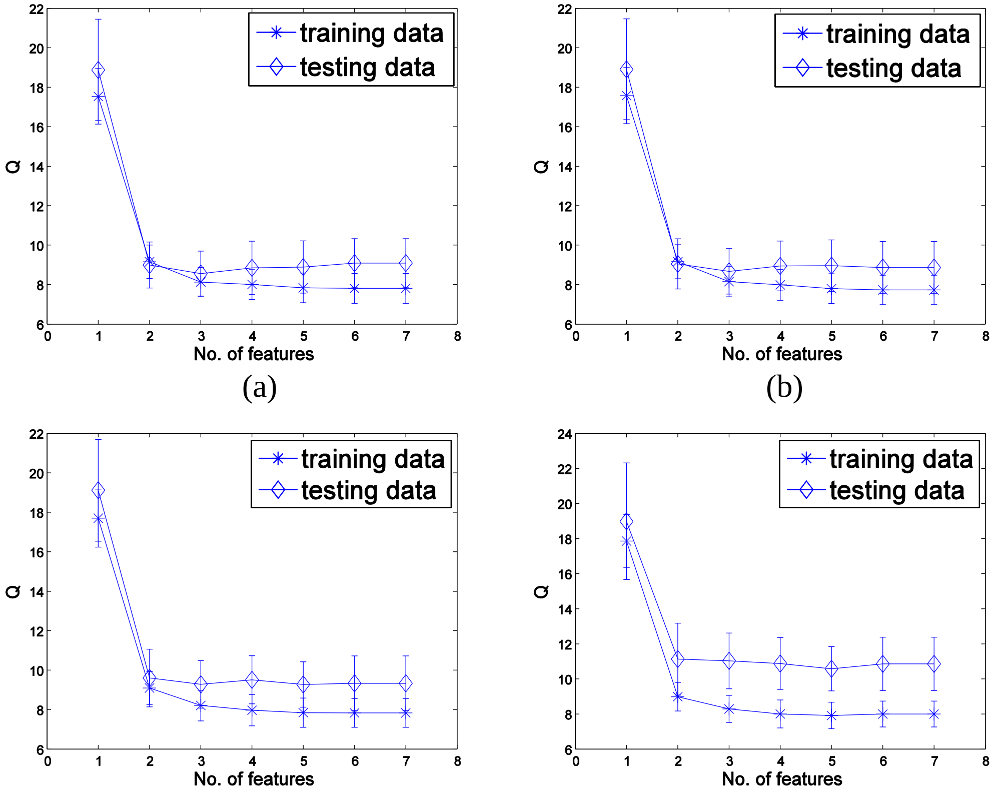
We present the results for several real data sets by focusing on the performance of the fuzzy models being treated as functions of the granularity of the architecture (c) and the dimensionality (p) of the reduced feature (input) space.

Table 5. Performance of the fuzzy model for the reduced feature space and optimized values of the fuzzification coefficients of the individual clusters

c features	2	3	4	5
First feature optimal value of ‘m”	0.0612 ±0.0022 0.1042 ±0.0090 [5.85 5.85]	0.0264 ±0.0008 0.0336 ±0.0033 [4.8 4.75 4.8]	0.0221 ±0.0009 0.0366 ±0.0108 [3.85 4.4 4.2 4]	0.0188 ±0.0007 0.0329 ±0.0141 [4.3 4.45 4.2 4.4 4.5]
Second feature optimal value of ‘m”	0.1878 ±0.0075 0.2002 ±0.0125 [1.8 1.9]	0.1740 ±0.0071 0.2403 ±0.0568 [4.35 4.1 4.55]	0.1632 ±0.0065 0.2114 ±0.0204 [3.8 4.2 3.15 3.3]	0.1601 ±0.0075 0.2921 ±0.0822 [4.15 4 4.35 4.5 4.05]

Auto MPG data set

In this data set, the automobile’s fuel consumption expressed in miles per gallon is to be the output of the model. The dataset includes 392 input–output pairs (after removing incomplete instances) where the input space involves 6.



(a)

(b)

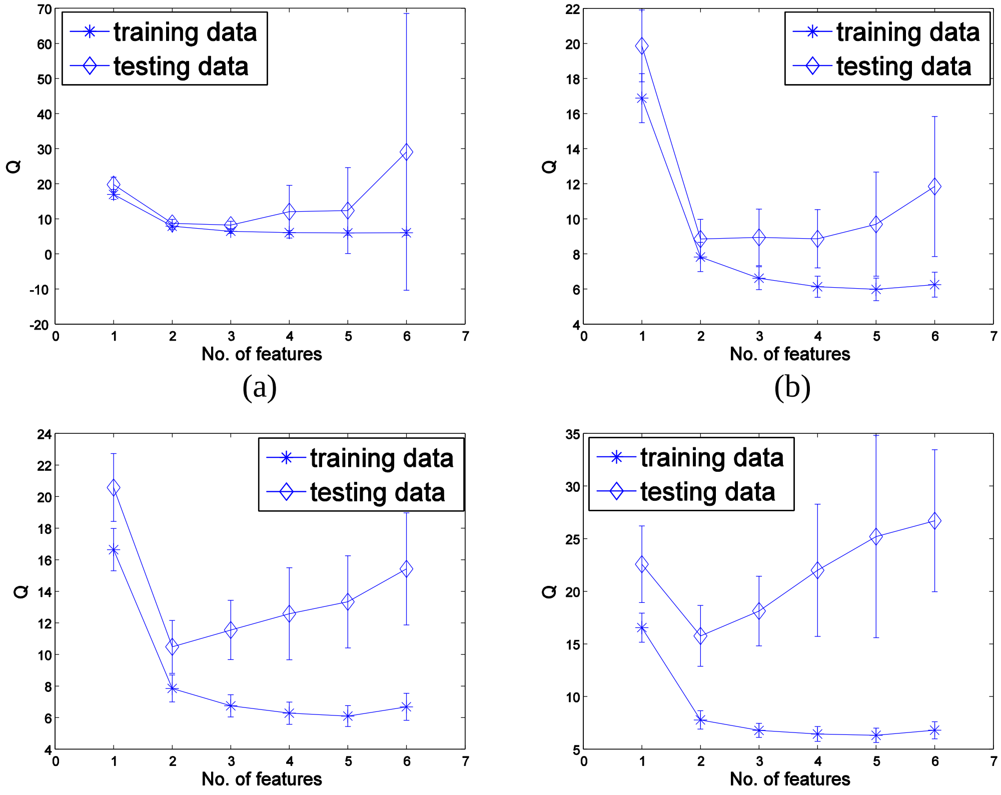
(c)

(d)

Figure 19. Performance index (mean value and standard deviation) versus the dimensionality of the successively reduced feature space for c = 2: (a) m = 1.5 (b) m = 2 (c) m = 3 (d) m = 4

10 fold cross-validation is adopted to quantify the performance achieved on the training and testing data. In each run, the data are subject to the 60% - 40% random split into the training and testing subsets. The results are visualized in Figure 19 where we show the values of the performance index for the training and testing data (both in terms of the mean value and standard deviation) for selected values of “m”.

It becomes apparent that the set of input variables can be significantly reduced by retaining only 2 or 3 inputs and this reduction is possible irrespective of the values of the fuzzification coefficient. The graphs shown in Figure 20 show the performance of the model for the increased level of granularity, that is c =5. Here an interesting tendency is clearly emphasized: with the increased dimensionality of the input space, the performance of the model on the training set is not very much improved however there is a clear deterioration of performance of the model when run on the testing data.



(a) (b)

(c) (d)

Figure 20. Performance index (mean value and standard deviation) versus the dimensionality of the successively reduced feature space for c = 5: (a) m = 1.5 (b) m = 2 (c) m = 3 (d) m = 4

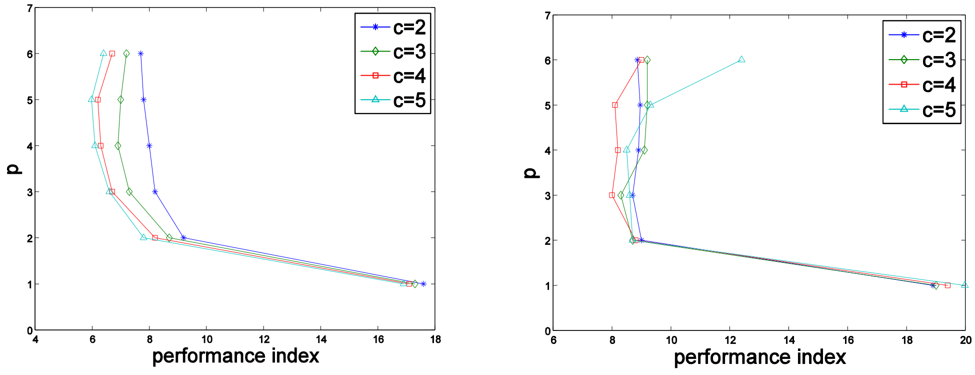
Table 6 summarizes the values of the performance index obtained for different levels of granularity (by varying c from 2 to 5 and keeping the same value of the fuzzification coefficient set to 2.0).

Table 6. Performance index Q versus the granularity of information (c) and the number of retained input variables (shown are the values mean values and standard deviations on the training and testing sets, m = 2)

p	1	2	3	4	5	6
2	17.5729 ± 1.4211 18.9065 ± 2.5516	9.1595 ±0.8620 9.0494 ±1.2683	8.1524 ±0.7720 8.6698 ±1.1510	7.9872 ± 0.7849 8.9437 ± 1.2623	7.7937 ± 0.7534 8.9619 ± 1.3059	7.7315 ± 0.7461 8.8648 ± 1.3229
3	17.3228 ± 1.5186 19.0289 ± 2.8038	8.6544 ±0.7201 8.7526 ±0.9740	7.2756 ±0.7161 8.3428 ±1.1507	6.9295 ±0.7663 9.0557 ±1.5033	7.0041 ±0.6128 9.1924 ±1.1062	7.2242 ±0.6786 9.2007 ±1.2870
4	17.1226 ±1.4717 19.4137 ± 2.8925	8.1999 ±0.8647 8.8479 ±1.1926	6.7196 ±0.6791 8.0306 ±1.0567	6.2978 ±0.6283 8.2109 ±1.0461	6.1886 ±0.5969 8.1364 ± 0.9766	6.6570 ±0.6446 9.0212 ±0.5723
5	16.9059 ± 1.3408 19.9814 ±2.0456	7.7949 ±0.7541 8.7357 ±1.3463	6.5609 ±0.7602 8.6393 ±1.1240	6.0883 ±0.6123 8.5388 ±1.0213	5.9816 ±0.5920 9.3210 ±1.6391	6.4183 ±0.6795 12.3718 ±3.7188

Figure 21 visualizes the results in the performance index – p coordinates. Here it becomes apparent that the changes in the values of the performance index are not affected too much when increasing the dimensionality of the input space. There is also a consistent pattern of behaviour with respect to the increased granularity of the space.

The results being reported in a tabular format in Table 7 focus on the details of the feature space by showing subsets of input variables (features), which form the reduced feature space; these relationships are reported for selected values of "m". The notation used here shows combinations of inputs which appear most often (results contained in the first brackets) which is followed by frequency it appeared (the result in the second bracket). For instance, for p = 3 and m = 2, the reduced subset of inputs is (3, 4, 6) and in the 10 fold cross validation it appeared 10 times out of 10 times.



(a) (b)

Figure 21. Visualization of the characteristics of the fuzzy model in the performance index – p coordinates. Shown are the relationships between the performance of the model and the increase of the dimensionality of the input space.

We can observe that the subsets of features are stable (viz. they appear quite consistently over all repetitions of the experiments) and the increase of the dimensionality of the input space results in adding new features while retaining the smaller subset that has been already identified. With this regard the growth of the input space results in the sequence of the inputs

p = 1 weight (4)
p = 2 weight (4), model year (6)
p = 3 weight (4), model year (6), horsepower (3)
p = 4 weight (4), model year (6), horsepower (3), displacement (2)
p = 5 weight (4), model year (6), horsepower (3), displacement (2), number of cylinders (1)

These subsets of inputs are intuitively appealing and reveal an interesting relationship between fuel consumption and the main characteristics of vehicles. If only a single input is to be considered then the weight comes into the play. For higher dimensionality of input spaces, the year of the model is to be considered and next horsepower and displacement start to appear in the realization of the model.

Table 7. Formation of reduced feature spaces and their stability (in terms of occurrence of subsets of features) for selected values of c and m

c = 2

p m	1	2	3	4	5
1.1	(4)(8/10)	(4,6) (10/10)	(3,4,6)(10/10)	(2,3,4,6)(10/10)	(1,2,3,4,6)(7/10)
2	(4)(9/10)	(4,6) (10/10)	(3,4,6)(10/10)	(1,3,4,6)(5/10)	(1,2,3,4,6)(6/10)
3	(4)(10/10)	(4,6) (10/10)	(3,4,6)(10/10)	(1,3,4,6)(9/10)	(1,2,3,4,6)(7/10)
4	(4)(10/10)	(4,6) (10/10)	(3,4,6)(10/10)	(1,3,4,6)(9/10)	(1,2,3,4,6)(9/10)

c = 3

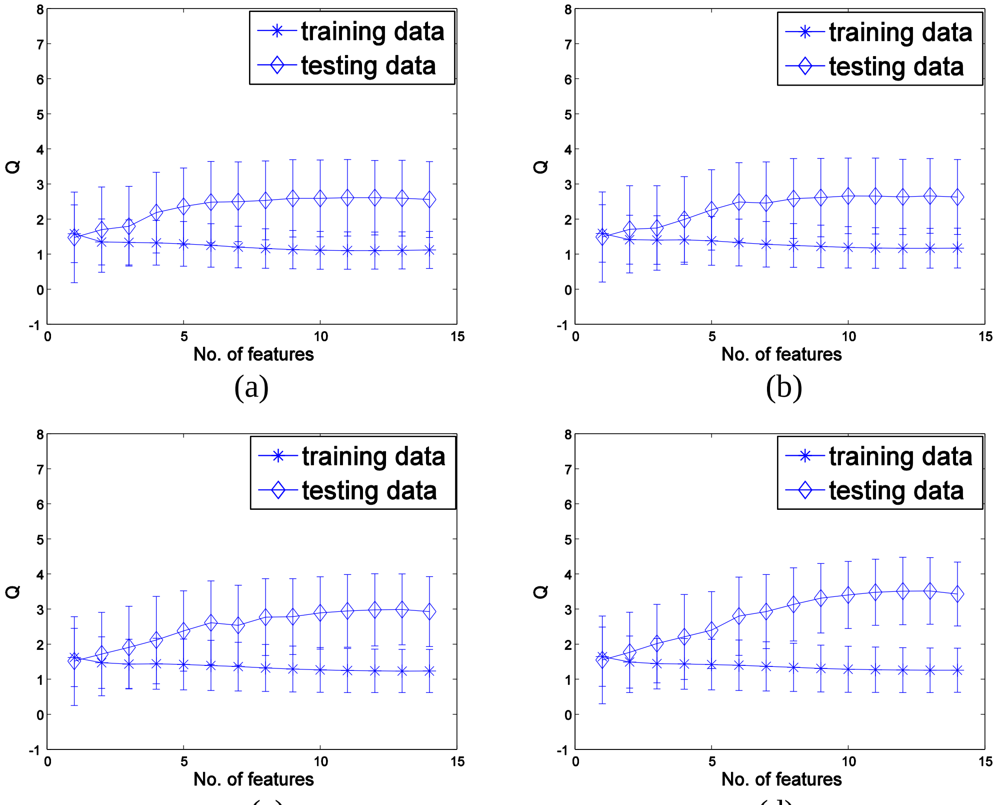
p m	1	2	3	4	5
1.5	(4)(6/10)	(4,6)(10/10)	(3,4,6)(10/10)	(1,3,4,6)(5/10)	(1,2,3,4,6)(4/10)
2	(4)(7/10)	(4,6)(10/10)	(3,4,6)(10/10)	(1,3,4,6)(8/10)	(1,2,3,4,6)(8/10)
3	(4)(6/10)	(4,6)(10/10)	(3,4,6)(10/10)	(1,3,4,6)(10/10)	(1,2,3,4,6)(9/10)
4	(4)(8/10)	(4,6)(10/10)	(3,4,6)(10/10)	(1,3,4,6)(9/10)	(1,2,3,4,6)(9/10)

c = 4

p m	1	2	3	4	5
1.5	(4)(6/10)	(4,6)(10/10)	(3,4,6)(10/10)	(1,3,4,6)(10/10)	(1,2,3,4,6)(5/10) (1,3,4,5,6)(5/10)
2	(4)(7/10)	(4,6) (10/10)	(3,4,6)(10/10)	(1,3,4,6)(10/10)	(1,2,3,4,6)(10/10)
3	(4)(7/10)	(4,6)(10/10)	(3,4,6)(10/10)	(1,3,4,6)(10/10)	(1,2,3,4,6)(10/10)
4	(4)(6/10)	(4,6)(10/10)	(3,4,6)(8/10)	(1,3,4,6)(9/10)	(1,2,3,4,6)(10/10)

Bodyfat data set

The Bodyfat data set (http://lib.stat.cmu.edu/datasets/bodyfat) is a collection of estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements for 252 men. The results are reported in Figure 22 and Table 8 and 9. It becomes apparent that for this data set, the feature space could be reduced to 1 or 2 features; the plot of Q is "flat" and does not change when reducing the number of the features.



(a)

(b)

(c)

(d)

Figure 22. Performance index versus the dimensionality of the feature space (c = 2): (a) m = 1.5 (b) m = 2 (c) m = 3 (d) m = 4

Table 8. Values of Q for selected values of p and c (m = 2)

p c	1	2	3	4	12	13	14
2	1.5888 ±0.8209 1.4870 ±1.2849	1.4106 ±0.6960 1.7082 ±1.2425	1.3848 ±0.6769 1.7665 ±1.0861	1.4152 ±0.7103 2.0012 ±1.1427	1.1652 ±0.5702 2.6298 ±1.0720	1.1665 ±0.5679 2.6562 ±1.0651	1.1719 ± 0.5668 2.6277 ± 1.0662
3	1.4661 ±0.7494 1.5105 ±1.2041	1.2570 ±0.6467 2.2371 ±1.2928	1.1584 ±0.5581 2.6145 ±1.1316	1.1949 ±0.6358 2.5815 ±1.2893	0.9271 ±0.4762 3.2899 ±0.9243	0.9327 ±0.4706 3.2643 ±0.9377	0.9479 ± 0.4702 3.2307 ± 0.8843
4	1.3540 ±0.6937	1.0306 ±0.4739	0.9140 ±0.4315	0.7449 ± 0.4801	0.4930 ±0.2811	0.6564 ±0.3185	0.7713 ±

	1.6698 ±1.1134	2.6551 ±1.2858	2.5282 ±1.1942	2.8897 ±1.3699	7.2003 ±5.1335	7.2523 ±3.5429	0.4001 5.8197 ± 3.0934
5	1.2430 ±0.6473 1.8746 ±1.1262	0.8648 ±0.4398 2.1813 ±1.2097	0.5905 ±0.3721 1.9842 ±1.3382	0.4689 ±0.3058 2.0719 ±1.5221	0.3207 ±0.2432 4.8568 ±3.7252	0.3345 ±0.2571 5.5248 ±3.6579	0.3942 ± 0.2984 9.3702 ± 9.5224

Table 9. Collections of features forming the reduced feature space for c = 2

m \ p	1	2	3	4	12	13
1.5	(1)(10/10)	(1,7) (8/10)	(1,7,11) (5/10)	(1,2,7,11) (4/10)	(1,2,4,5,6,7,9,10,11,12,13,14) (2/10)	(1,2,4,5,6,7,8,9,10,11,12,13,14) (3/10)
2	(1)(10/10)	(1,7) (7/10)	(1,7,11) (6/10)	(1,2,7,11) (5/10)	(1,2,3,5,6,7,8,9,10,11,12,13) (3/10)	(1,2,3,5,6,7,8,9,10,11,12,13,14) (4/10)
3	(1)(10/10)	(1,7) (7/10)	(1,7,11) (5/10)	(1,2,7,11) (5/10)	(1,2,3,5,6,7,8,9,10,11,12,13) (2/10)	(1,2,3,5,6,7,8,9,10,11,12,13,14) (5/10)
4	(1)(10/10)	(1,7) (7/10)	(1,7,11) (5/10)	(1,2,7,11) (5/10)	(1,2,3,5,6,8,9,10,11,12,13,14) (2/10)	(1,2,3,5,6,7,8,9,10,11,12,13,14) (4/10)

The reduced input space is formed by the following features (here c = 2)

p = 1; Percent body fat from Siri's (1956) equation (1)
p = 2: Percent body fat from Siri's (1956) equation (1), Abdomen 2 circumference (7)
p = 3: Percent body fat from Siri's (1956) equation (1), Abdomen 2 circumference (7), Ankle circumference (11)

5.4. Conclusions

The design of fuzzy models inherently invokes the fundamentals and the development of information granulation. Among other factors, the interpretability of the fuzzy model relates to the dimensionality of the input space and thus its reduction becomes one of the efficient ways of increasing the transparency of the model. The reduction problem is of combinatorial nature, in which we can resort ourselves to the methods of Evolutionary Computing and swarm optimization.

Through a series of experiments, we showed that the reduction of the input space becomes possible and the reduced space is fairly stable (viz. consists of the same subsets of inputs) irrespective of the varying granularity of the model. The level of reduction varied from data to data and this could have been anticipated. In some cases it was shown that the reduced space led to better performance of the models in terms of the resulting accuracy. We also demonstrated that the optimization of the fuzzification coefficient impacts the quality of the model. Furthermore the fuzzification coefficient relates to the interpretability of the rules considering that the level overlap between information granules translates into the level of interaction between the rules.

Further investigations can include nonlinear local models in which case we may envision the ability to use a lower number of rules (local models)or other modeling architectures like neural networks. The long-term research agenda worth pursuing may embrace a general concept of modeling resources such as the overall number of subconditions, order of local models (treated as polynomials) which are to be allocated throughout all the rules so that a predefined objective function becomes minimized. This positions us in the realm of structural optimization with the anticipated use of the machinery of evolutionary and population-based computing.

6. Granular Fuzzy Modeling

We have proposed a new feature reduction method in fuzzy modeling and applied it in linear fuzzy modeling. Both feature reduction and linear fuzzy models are fundamental considerations when we do system modeling. In this chapter, we propose an advanced fuzzy model: a hierarchy fuzzy model aggregated by local fuzzy models. A new aggregation technique is proposed here in which each local model is assigned with a granularity. The result comes in the form of a so called granular model, which directly reflects upon and quantifies the diversity of the available sources of knowledge (local models) involved in knowledge management. Several detailed algorithmic schemes are presented along with related computational aspects associated with Granular Computing.

It is also shown how the construction of information granules completed through the use of the principle of justifiable granularity becomes advantageous in the realization of granular fuzzy models and a quantification of the quality (specificity) of the results of modeling. We focus on the design of granular fuzzy models considering that the locally available models are those fuzzy rule-based. Numeric results are discussed with intent of displaying the most essential features of the proposed methodology and algorithmic developments.

6.1. Ensemble of Models

In system modeling, knowledge management comes vividly into the picture when dealing with a collection of individual models. These models being considered as sources of knowledge are engaged in some collective pursuits of a collaborative development to establish modeling outcomes of global character. We here propose a new method thus that the final result comes in the form of a so-called granular fuzzy model, which directly reflects upon and quantifies the diversity of the available sources of knowledge (local models) involved in knowledge management.

We enrich the plethora diversity of existing methodologies and algorithms by moving from single, individual and local models to a family of models in which individual models collaborate with others under an ultimate intent of forming a more abstract, holistic and general model of the underlying phenomenon, process or a system delivering a global albeit less detailed view at the reality. This shift has brought a number of new challenges along with tangible benefits and better rapport with reality irrespectively from the development technologies. There is no surprise that in fuzzy modeling with its plethora of design techniques, see (Alcala, et al. 2009) (Chang and Wang 2009) (Delgado, Gomez-Skarmeta and Martin 1997) (Gobi and Pedrycz 2007) (Kacprzyk and Zadrozny 2001) (Liang and Pedrycz 2009) (Park, Pedrycz and Oh 2009) (Roubos and Setnes 2001) involving criteria of accuracy and interpretability (Eftekhari, et al. 2008) (Ishibuchi and Nojima, Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning 2007) and invoking promising methods of global optimization (Pham and Castellani 2006), this concept has to translate into sound concepts, methodology, design strategies, and finally detailed algorithms.

It is worth noting that there have been some developments along the line of collaboration, consensus–building and knowledge exchange in decision-making (Pasi and Yager 2006) (Herrera, Martinez and Sanchez, Managing non-homogeneous information in group decision making 2005) (Pedrycz and Rai, A multifacted perspective at data analysis: a study in collaborative intelligent agents 2008); however the role and unavoidable emergence of information granules as a vehicle to quantify a diversity of sources of knowledge has not been fully exploited. Let us observe the general character of a system which helps motivate this study.

Consider a system for which formed is a series of models. This system can be perceived from different, quite diversified points of view. It can be observed over some time periods, some different locations and analyzed at different levels of details. Subsequently, each local model is built with a certain objective in mind. It is a very interesting scheme to form a holistic model of the system by taking advantage of the individual sources of knowledge which has been constructed so far. When doing this, it is obvious that the different sources of knowledge show a diversity character and hence this diversity has to be taken into consideration and quantified carefully. No matter what features individual models own, it is legitimate to anticipate that the global model (at the higher level of a hierarchy) is more abstract, general. Due to the nature of information granules (Pedrycz, Knowledge-Based Clustering: From Data to Information Granules 2005) (Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic 1997) (Zadeh, Toward a generalized theory of uncertainty (GTU) — an outline 2005) (Zadrozny and Kacprzyk 2006) (generation of numeric information), it is legitimate to assume that there is a sort of formalism of information granules in global models. Thus information granules offer a unique way to quantify the diversity of sources of knowledge and express this in the way of the level of granularity.

From the reviewed literature (Yager, On ordered weighted averaging aggregation operators in multicriteria decision making 1988) (Yager, Hierarchical aggregation functions generated from belief structures 2000) (Wu and Mendel 2007) (Chen, Tiho and Yao 2009) (Zhou and Yu 2005) (Kuncheva and Rodriguez 2007) (Islam, et al. 2008) (Reformat, A fuzzy-based multimodel system for reasoning about the number of software defects 2005), we can find that the concept of using a collection of models has been around for some time. The term ensemble is commonly used to describe the combined model. It can be demonstrated from compelling evidence that the performance of the ensemble is substantially better than a single model. We provide a list of existing approaches which are most popular in Table 10.

In spite of the visible diversity present in the table, there is also a clear similarity: the result of aggregation is positioned at the same level of granularity as the results produced by the individual models involved in the aggregation.

Table 10. Selected models and their aggregation

Reference	Local models	Aggregation mechanism	Aggregation result
(Yager, On ordered weighted averaging aggregation operators in multicriteria decision making 1988)	"n" numeric values in [0, 1] of "n" criteria {A_1, A_2, ..., A_n},	OWA (ordered weighted averaging):	numeric value
(Yager, Hierarchical aggregation functions generated from belief structures 2000)	"n" numeric values in [0, 1] of "n" criteria {A_1, A_2, ..., A_n},	$G(a_1,a_2,...,a_n) = \sum_{i=1}^{q} \alpha_i Fw_i([B_i])$ B_i is a subset of relevant criteria, M_i provides a description of what portion of the criteria in need be satisfied in order for the module to be satisfied and α_i indicates the value of satisfying this module	numeric value
(Wu and Mendel 2007)	"n" interval-valued fuzzy sets	$Y = \frac{\sum_{i=1}^{m} w_i x_i}{\sum_{i=1}^{m} w_i}$	interval –valued fuzzy set
(Chen, Tiho and Yao 2009)	n CART trees in an ensemble	EP (Expectation propagation) pruning algorithm	a subset of ensemble members (bagging, Adaboost, and Random Forest used to generate ensemble)
(Zhou and Yu 2005)	n nearest-neighbor classifiers	majority voting	numeric class label
(Fumera and Roli 2005)	"n" MLP classifiers	Linear function combiner with optimal values of the classifier weights	numeric class label
(Kuncheva and Rodriguez 2007)	"n" decision trees	Each classifier in the ensemble is replaced by a miniensemble of a pair of subclassifiers with a random linear oracle to choose between the two.	an ensemble; numeric output

(Islam, et al. 2008)	"n" neural network (NN) classifiers	No aggregation; each NN is an element of the ensemble; the smallest error on a validation set.	an ensemble of "n" neural networks-numeric output

6.2. Formation of Granular Fuzzy Models

As explicitly described in chapter 3, the principle of justifiable granularity is concerned with a formation of a meaningful representation of a collection of numeric values. It is clear that the component of granularity brings a desired level of flexibility required to reflect and quantify an inherent diversity of individual fuzzy models. The underlying concept is concisely illustrated in Figure 23. The unavoidable granularity effect manifests at the resulting model formed at the higher level of abstraction, hence the construct obtained there comes as a generation of fuzzy models present at the lower level of the hierarchy, namely granular fuzzy models.

A certain system or phenomenon is perceived from different points of view (perspectives). The fuzzy models emerging there, denoted as FM-1, FM-2, …, FM-p use locally available data D-1, D-2, …, D-p. In general, we can envision fuzzy models to be of different nature (say, rule-based, neurofuzzy model, fuzzy cognitive map); however for the sake of the clarity of overall presentation we assume that all of them are fuzzy rule-based models here. The models FM-1, FM-2, …, FM-p are now brought together by forming a global model at the higher level of hierarchy. The model at the higher level is definitely more abstract (general) than those at the lower level as it tries to embrace a variety of sources of knowledge (fuzzy models). As such we envision that the output of such fuzzy model is less specific than the numeric output generated by any of the fuzzy models at the lower level. Imagine that the input is equal to **x** with **x** being a numeric vector of inputs coming from the system. Each fuzzy model produces the numeric outputs FM-1(**x**), FM-2(**x**), …., FM-p(**x**).

In virtue of the different perception of the system itself, it is very likely that these outputs are different; we do not rule out that they could be close to each other. If the fuzzy model at the higher level is to capture this diversity of the sources of knowledge (viz. fuzzy models), intuitively for the same **x** it should return an output that is more abstract than a single numeric entity; that is a certain information granule. Hence such fuzzy models producing granular outputs will be referred to as *granular* fuzzy models.

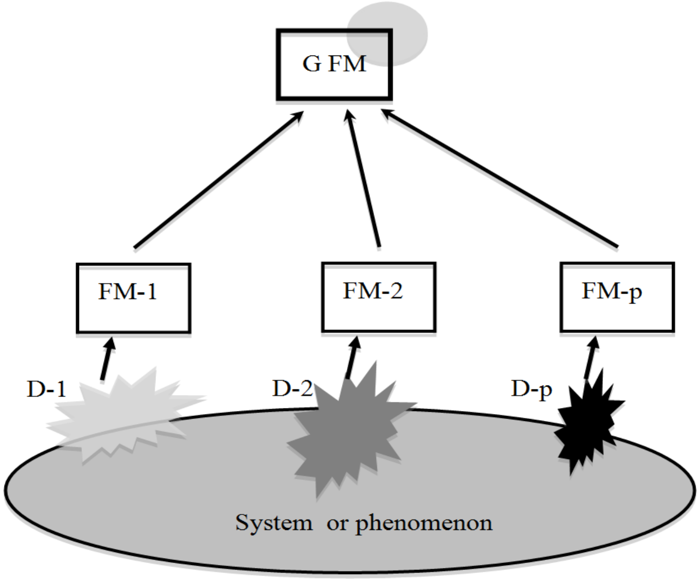


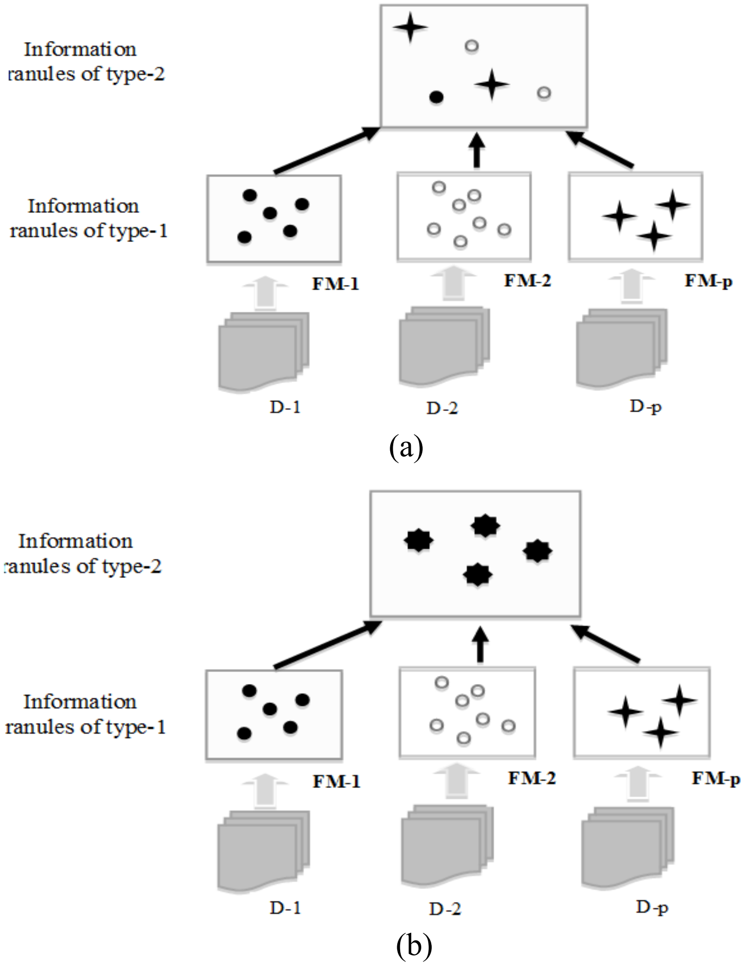
Figure 23. The emergence of information granularity in the model at the upper level of the hierarchy as a result of dealing with the diversity of fuzzy models FM-1, FM-2, …, FM-p. The emergence of information granularity is denoted by large grey dot placed next to the granular fuzzy model being formed there

Alluding to Table 10, we can contrast the proposed construct with the approaches existing in the literature, especially that on a surface there seem to be some similarities. There are several evident differences:

- The result of aggregation is positioned at the higher level of abstraction in comparison with the results generated by the individual models. For instance, the models involved in the aggregation produce numeric outputs and numeric outputs are also generated as a result of aggregation. Here, the outputs are granular (coming in the form of intervals, fuzzy sets, rough sets, etc.) even though the individual models generate numeric outputs. This can be viewed as a substantial advantage of the overall topology as it helps quantify the diversity of the individual sources of knowledge (viz. models). The quantification of this diversity is realized by running the principle of justifiable granularity.
- The data are not shared meaning that each model is constructed on a basis of locally available data; furthermore the variables at each data set need not be the same. This stands in sharp contrast with the scenario of bagging or boosting of a family of models.
- The individual models could be very different (say, neural networks, rule–based models, linear regression, etc.)

In the following, we present in more detail one scenario of capturing sources of knowledge (fuzzy models) in case the individual sources of knowledge-fuzzy models are treated as a collection of fuzzy rule-based models. The transfer of knowledge realized here is focused on the use of information granules forming a backbone of the individual fuzzy models at the lower level. As the antecedents of the rules are formed on a basis of information granules, a realization of a certain model at the higher level of hierarchy calls for a formation of a collection of information granules to start with. Here we envision two main directions in the construction of information granules as being portrayed in Figure 24:

(a) A selection of a suitable subset of information granules forming the individual models, Figure 24 (a). The prototypes of the information granules are selected in such a way so that they represent all prototypes of the models to the highest extent. This is a combinatorial optimization problem, which may call for techniques of Evolutionary Optimization or population-based optimization to arrive at solutions to the problem. The optimization criterion quantifies a reconstruction error of the prototypes at the lower level when being expressed in terms of the subset of the prototypes formed at the upper level.



(a)

(b)

Figure 24. Formation of the information granules at the higher level: (a) selection and (b) clustering of prototypes

(b) The second approach, illustrated in Figure 24 (b), is concerned with clustering (granulation) of prototypes available at the lower level. The standard Fuzzy C-Means (FCM) can be used here. It operates on a family of prototypes present in all fuzzy models positioned at the lower level of hierarchy and produces "c" prototypes at the upper level of the hierarchy. In light of the construction of the information granules, which have been built at the higher level (which are also sought as more abstract view at the information granules present at the lower level), we may refer them as information granules of higher type, say type-2 information granules (and type-2 fuzzy sets, in particular).

Given the collection of information granules (which can be represented as a family of the prototypes), we are at position to develop a model at the higher level. We note that there is an inherent granularity of the associated model, which comes from the fact that for any prototype formed at the higher level $\mathbf{v}_i$, each fuzzy model at the lower level returns some numeric value, say, FM-1($\mathbf{v}$1), FM-2($\mathbf{v}_2$),…, FM-p($\mathbf{v}_i$), see Figure 25. It is very unlikely that all these values are the same. This set of data is subject to the granulation process (with $\mathcal{G}$ denoting a granulation mechanism). The size (level of granularity) of the resulting information granule depends upon the predetermined value of the parameter α, which was used to construct information granules. We will be taking advantage of this flexibility in the realization of the model guided by two conflicting objectives (as this will be discussed later in more detail).

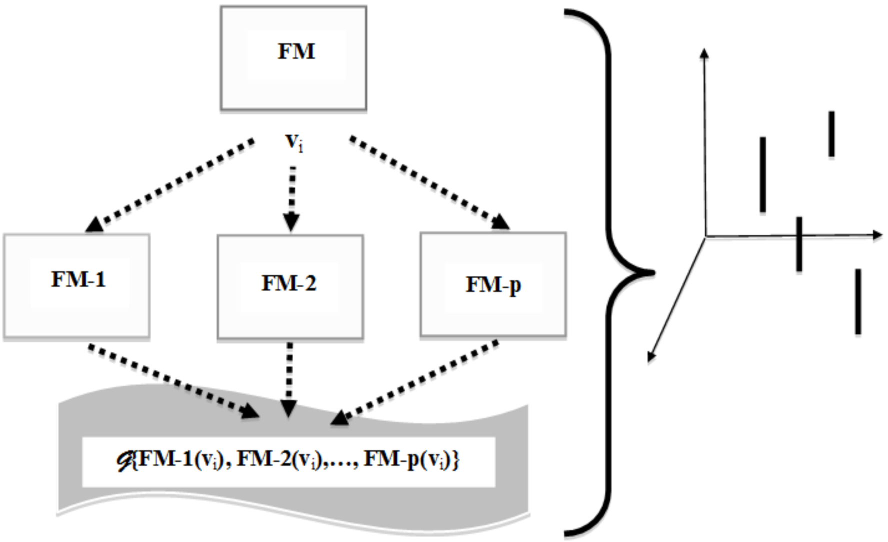

FM
$\mathbf{v}_i$
FM-1
FM-2
FM-p
$\mathcal{G}$\{FM-1($\mathbf{v}_i$), FM-2($\mathbf{v}_i$),…, FM-p($\mathbf{v}_i$)\}

Figure 25. Experimental evidence behind the formation of the granular model to be constructed at the higher level, considered here is an interval format of resulting information granules

Overall, for "c" information granules built at the higher level, the available experimental evidence (originating at the lower level of hierarchy) arises in the form

$$
\begin{aligned}
&\{\, (\mathbf{v}_1 , \mathcal{G}(\text{FM-1}(\mathbf{v}_1), \text{FM-2}(\mathbf{v}_1), ..,\text{FM-p}(\mathbf{v}_1))),\\
&(\mathbf{v}_2 , \mathcal{G}(\text{FM-1}(\mathbf{v}_2), \text{FM-2}(\mathbf{v}_2), \ \ldots, \text{FM-p}(\mathbf{v}_2))), \quad \ldots\\
&(\mathbf{v}_i , \mathcal{G}(\text{FM-1}(\mathbf{v}_i), \text{FM-2}(\mathbf{v}_i), \ \ldots, \text{FM-p}(\mathbf{v}_i))), \ \ldots\\
&(\mathbf{v}_c , \mathcal{G}(\text{FM-1}(\mathbf{v}_c), \text{FM-2}(\mathbf{v}_c), \ \ldots, \text{FM-p}(\mathbf{v}_c))) \,\}
\end{aligned}
\qquad (6.1)
$$

and invokes an evident component of granularity as highlighted in Figure 25. Taking these data into consideration, we construct a granular model. Preferably the model of this nature has to be structure free as much as possible. In this case, a technique of case-based reasoning (CBR), or being more specific, granular case based reasoning can be anticipated. The other option worth considering is a concept of fuzzy regression.

6.3. Computation of Granular Outputs

Moving to the computational details, we compute a degree of activation of the cases by a certain input **x**.

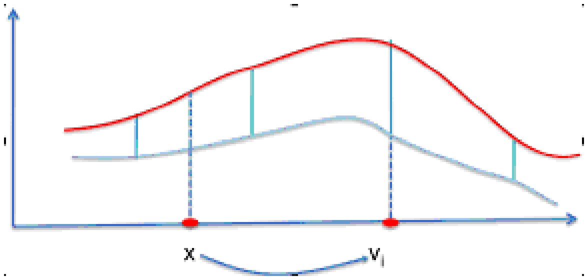


Figure 26. The mechanism of granular case-based reasoning and a construction of lower and upper bounds

$$
u_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^{c} \left(\frac{\|\mathbf{x} - \mathbf{v}_i\|}{\|\mathbf{x} - v_j\|} \right)^{2/(m-1)}}, \quad m>1 \qquad (6.2)
$$

As the outputs are evidently granular, we determine the lower and upper bounds of the granular model based on the bounds of y_i^- and y_i^+ where $[y_i^- , y_i^+]= \mathcal{G}(\text{FM-1}(\mathbf{v}_i), \text{FM-2}(\mathbf{v}_i), ..,\text{FM-p}(\mathbf{v}_i))$. We obtain the following expressions for the bounds:
lower bound

$$
y^-(\mathbf{x}) = \sum_{i=1}^{c} u_i(\mathbf{x}) y_i^- \qquad (6.3)
$$

upper bound

$$y^{+}(\mathbf{x}) = \sum_{i=1}^{c} u_i(\mathbf{x}) y_i^{+} \tag{6.4}$$

An example of the granular "envelope" produced in this way is shown in Figure 27. Note that the parameter "m" (fuzzification coefficient) affects the shape of the granular mapping (and this could be used as an additional parametric flexibility inbuilt into the granular model).

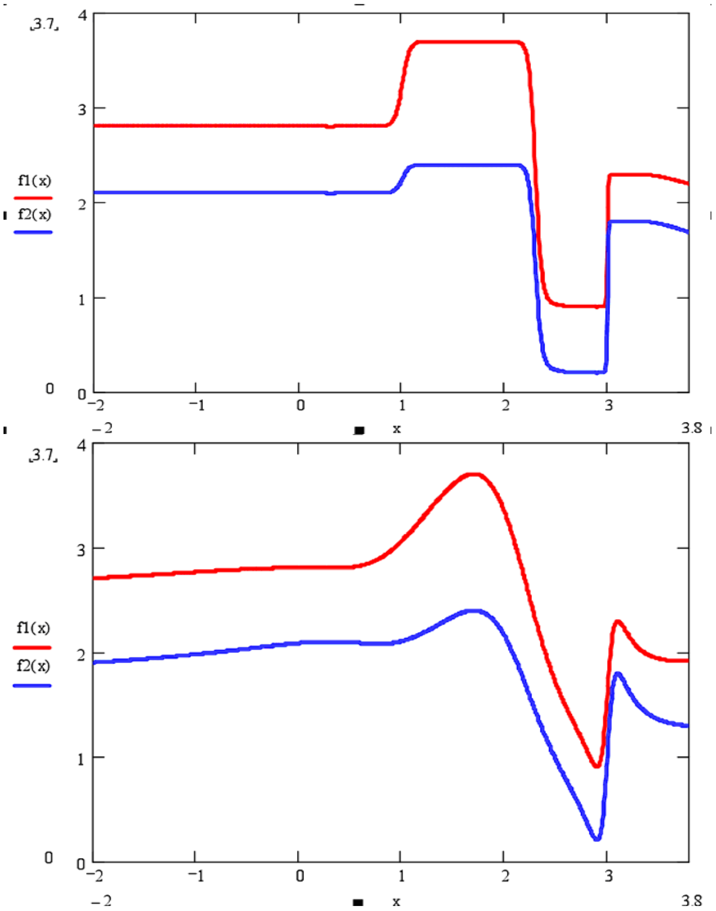


Figure 27. Plots of the granular model for selected values of m; m=1.2, 2.0

The granularity of the data themselves depends upon the value of α being used when running the process of justifiable granularity. We are faced with a two-objective optimization problem where one of the objectives is to make the envelope as narrow as possible (to achieve high specificity level, which is desirable) and at the same "cover" as much experimental evidence as possible (so a large number of data are included within the lower and upper bound making the granular model highly legitimized by the experimental evidence captured by the series of fuzzy models at the lower level of the hierarchy). Intuitively, the two requirements, which are evidently in conflict, are illustrated in Figure 28.

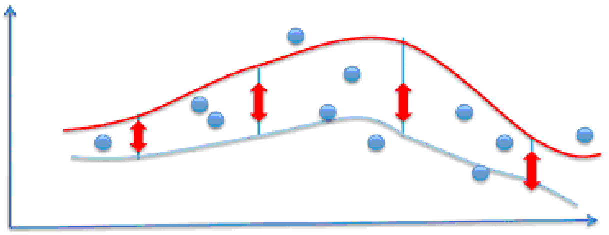


Figure 28. Illustration of the two objectives to be optimized in the construction of the granular model

The satisfaction of the two requirements and a way of achieving a certain compromise can be controlled by choosing a certain value of information granularity level α.

6.4. Experimental Studies

In a series of numeric experiments, we present a design and performance of granular fuzzy models and highlight some essential features of the models along with a role played by the key design parameters of the granular constructs formed at the upper level of the hierarchy. In all experiments, the value of the fuzzification coefficient was set to 2.0 or was subject to some optimization.

Concrete compressive strength data

This data set coming from the Machine Learning repository (http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength) concerns a compressive strength of concrete with association with its characteristics such as blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate. It consists of 1,030 pairs of input –output data. We form 10 fuzzy rule-based models where each of them is constructed on a basis of 103 randomly selected data. The number of rules in each model is equal to the number of fuzzy clusters determined for each locally available data set. The performance index Q, which is used to evaluate the quality of the model is expressed in (2.12).

Once the information granules have been formed by using the FCM algorithm, the parameters of the linear functions present in the conclusion parts of the rules were estimated by running a standard least squares estimation (LSE) method. The number of rules itself is subject to optimization and here we resort ourselves to a successive enumeration by designing the fuzzy model for increasing the number of rules while monitoring the values of the corresponding performance index (6.5). The values of Q treated as a function of the number of rules (clusters) for all the models are shown in Figure 29.

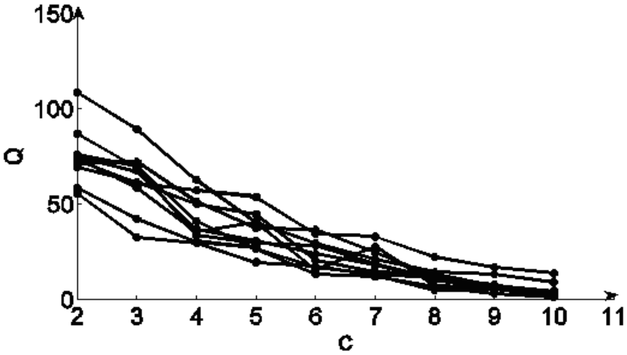


Figure 29. The performance index of the models versus the number of rules (clusters); shown are the values for all 10 models

One can note (as it has been expected) that when the number of clusters is increased, the performance index decreases. Essentially all the local models follow the same tendency: there is a cutoff point at c = 5 or 6 beyond which the values of the performance index are not significantly affected by further increasing the number of rules. To visualize the performance of the fuzzy models formed here, a collection of plots of output data versus the corresponding outputs of the models is included in Figure 30.

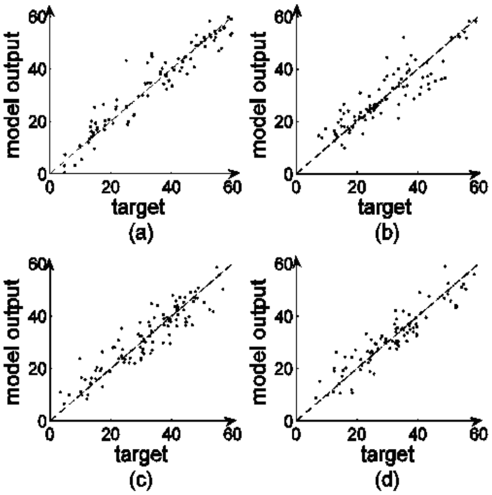


Figure 30. The performance of selected fuzzy models (data versus model output): (a) Q = 21.94 (b) Q = 31.16 (c) Q = 29.25 (d) Q = 24.01

Once the 10 local models have been constructed, the prototypes present there are clustered at the higher level to create a backbone of the global fuzzy model. Following the principle of justifiable granularity, for each of the prototypes obtained in this way we form the intervals of the corresponding output values coming from all local models. Several selected information granules are illustrated in Figures 31-32 where the number of clusters built at the higher level is set to c = 3 and 10. The values of the granularity level α are set to quite low and quite high values; this helps contrast the specificity of the resulting intervals (indicated in the figure by different levels of shading).

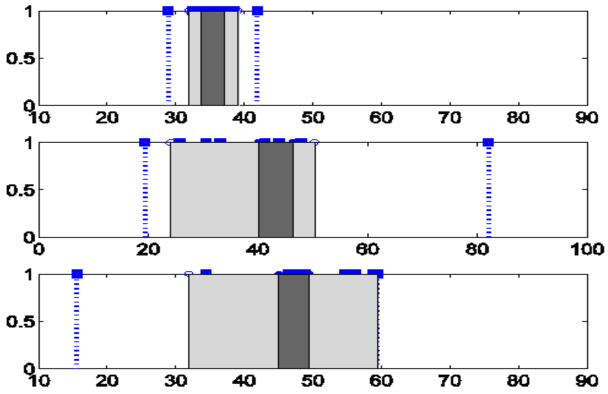


Figure 31. Prototypes and the associated intervals for two values of α (0.01 and 0.76), the number of clusters (c) is set to 3.

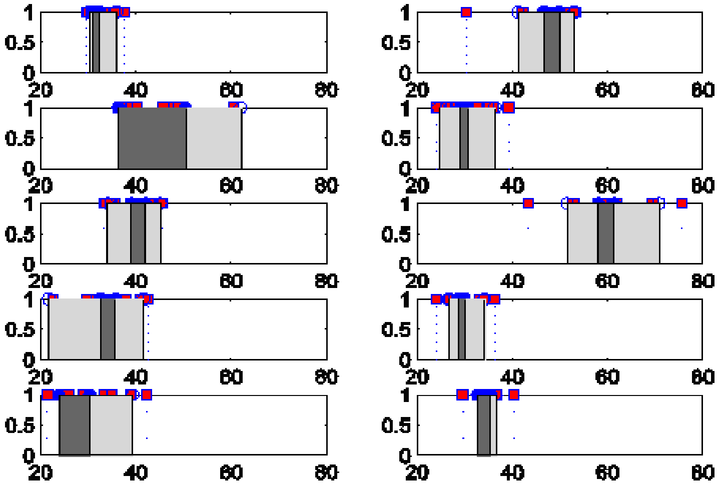


Figure 32. Prototypes and the associated intervals for two values of α (0.01 and 0.71), the number of clusters is set to 10.

The most concise and informative characterization of the quality of the overall model can be conveyed by plotting the values of the coverage of data “I” versus the cumulative length of the intervals produced by the granular fuzzy model for the inputs formed as the prototypes of the information granules obtained for the individual fuzzy models. The resulting relationships are shown in Figure 33.

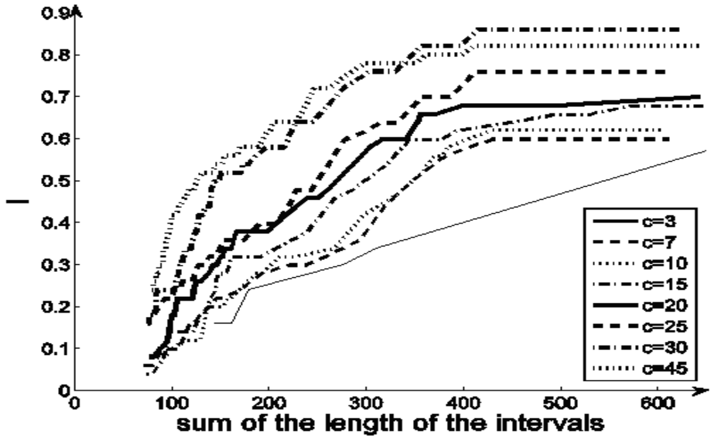


Figure 33. I versus the cumulative length of intervals L of the output for selected number of clusters

It becomes apparent that “I” is a non-decreasing function of the cumulative length of the intervals. Furthermore the index “I” assumes higher values for increasing values of “c”. The values of AUC are determined by computing the area of the plots presented in the (3-α, I) coordinates. As noted earlier, we use another x-coordinate (that is 3-α) to produce consistent results for all combinations of the parameters. This is not always possible when using the cumulative length of the intervals. For instance, we observe this effect in Figure 33 where the curve for c = 3 starts at higher values of L. The choice of the new variable 3-α is motivated by the fact that for values of higher than 3 there is a saturation effect in the sense no substantial changes in the values of I are reported. The corresponding series of plots is shown in Figure 34. We report the values of I for m =2 (which is the typical value of the fuzzification coefficient in the FCM method) as well as for the optimal values of m. In all cases reported here, there is a visible difference (improvement) when using the optimized value of “m”. As a matter of fact, the optimal values of the fuzzification coefficient vary in-between 1.5 and 6.0. There is also a similar relationship between the values 3-α and the coverage levels which increase quickly when α assumes low values (and the intervals become wider).

The plots of the outputs from the data versus the outputs (intervals) of the granular fuzzy model are presented in Figures 35 and 36. Here we contrast the lengths of the intervals and their position when using a different number of clusters (c was set to 3 and 40, respectively).

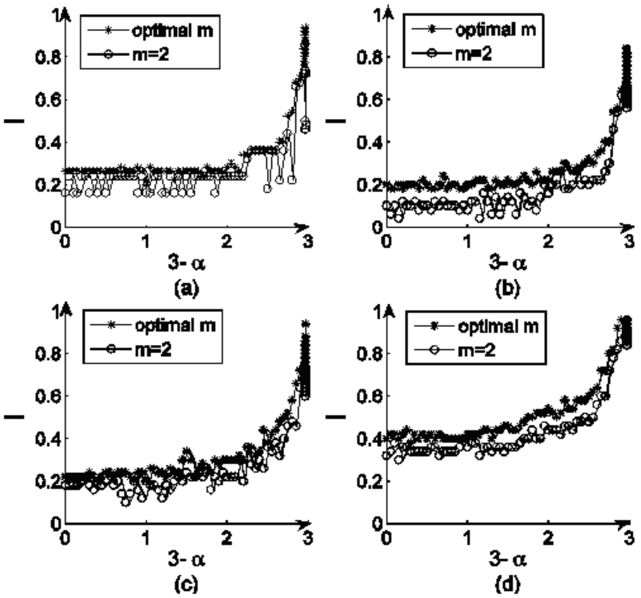


Figure 34. I versus the level of granularity, 3-α for selected values of c: (a) c=3, (b) c=10, (c) c=20, (d) c=40.

As could have been expected, the specificity of output information granules increases with the larger number of rules (clusters) used in the granular fuzzy model. Table 11 summarizes the values of the AUC for different numbers of "c" and the two strategies of forming information granules at the higher level, which is clustering the existing prototypes and the selection of their subsets. The fuzzification coefficient is also optimized.

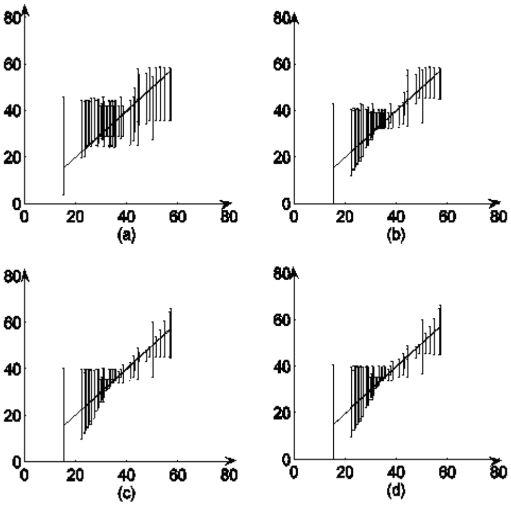


Figure 35. Scatter plots of data versus the intervals generated by the corresponding inputs obtained for different values of α: (a) α = 0.01, (b) α = 0.1, (c) α = 1, (d) α = 3, c = 3.

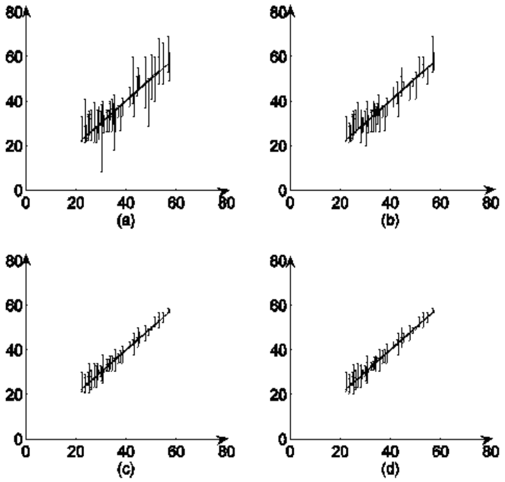


Figure 36. Scatter plot of data versus the intervals generated by the corresponding inputs obtained for different values of α: (a) α = 0.01, (b) α = 0.1, (c) α = 1, (d) α = 3, c = 40.

Table 11. Values of the AUC for selected numbers of clusters and two strategies of the determination of the prototypes of the granular fuzzy model

Strategy of prototype formation	c = 3		c = 10		c= 20		c = 40	
	Optimal m	m = 2	Optimal m	m = 2	Optimal m	m = 2	Optimal m	m = 2
clustering	0.9271	0.7682	0.7597	0.4902	0.9184	0.7271	1.5105	1.2774
selection	0.7680	0.2579	0.7846	0.4809	0.8326	0.5736	0.9558	0.6450

Two observations are well supported by the obtained results. In general, the quality of the granular model improves with the increase in the number of clusters. This becomes for both selection strategies (clustering of prototypes or their selection). The clustering of the prototypes is a better strategy leading to the higher values of the AUC. There is also a beneficial effect of the optimization of the fuzzification coefficient – in all cases we report the improvement, which is more visible when associated with the selection strategy.

When we use different number of clusters (rules) in the models formed at the lower (more specifically, 6, 5, 6, 5, 4, 6, 4, 4, 5, 6), the performance of the granular fuzzy model is illustrated in Figure 37.

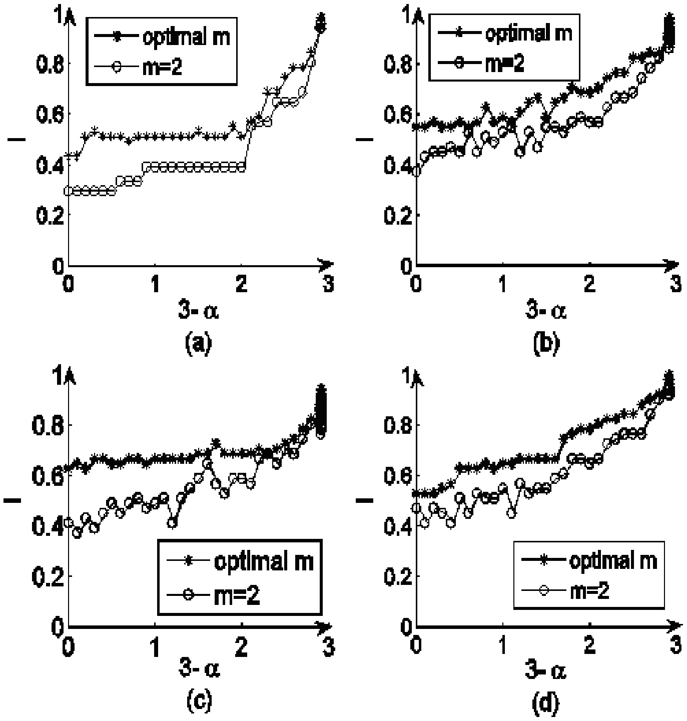


Figure 37. I versus the level of granularity, 3-α for selected values of c: (a) c=3, (b) c=10, (c) c=20, (d) c=40. Here we use the strategy of prototype clustering.

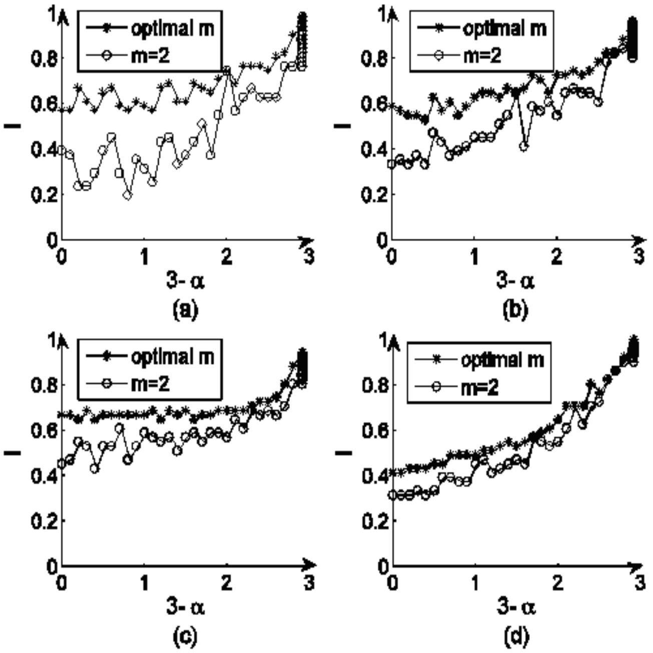


Figure 38. I versus the level of granularity, 3-α for selected values of c: (a) c=3, (b) c=10, (c) c=20, (d) c=40.

In case of using the strategy of selecting prototypes (out of the prototypes available at the lower level), the obtained results are displayed in Figure 38. Here the optimal values of "m" are reported to be located in the range of 1.5- 6. The values of AUC for selected numbers of clusters are list in Table 12.

Table 12. Values of the AUC for selected numbers of clusters and two strategies of the determination of the prototypes of the granular fuzzy model

strategy of prototype formation	c = 3		c = 10		c= 20		c = 40	
	Optimal m	m = 2	Optimal m	m = 2	Optimal m	m = 2	Optimal m	m = 2
clustering	1.6741	1.3099	1.9393	1.6502	2.0171	1.6353	2.0832	1.7697
selection	1.9867	1.3678	1.9668	1.5696	2.0300	1.7174	1.7331	1.5247

PM10 data set

This data set comes from StatLib (http://lib.stat.cmu.edu/datasets/) and comprises 500 observations originating in a study where air pollution at a road is related to traffic volume and meteorological variables, collected by the Norwegian Public Roads Administration. The number of input variables is equal to 7. The overall data set is divided into 10 equal size subsets, which were used to construct 10 local fuzzy models at the lower level. The performance of these models reported for selected values of "c" ranging from 2 to 6 is illustrated in Figure 39. The number of clusters used in each local model is set to 4.

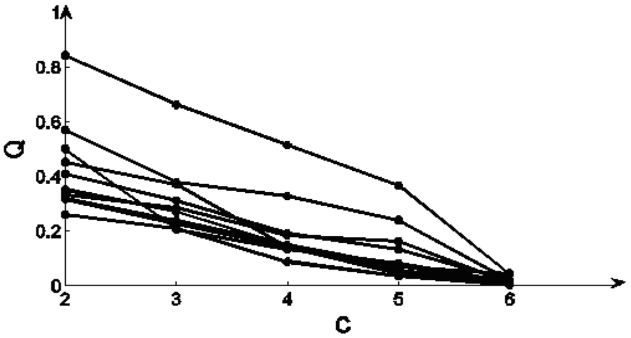


Figure 39. The performance index Q versus the number of rules (clusters); m=2.0

The performance of the local models is quantified by plotting the outputs of the corresponding models versus the target; the series of plots is provided in Figure 40.

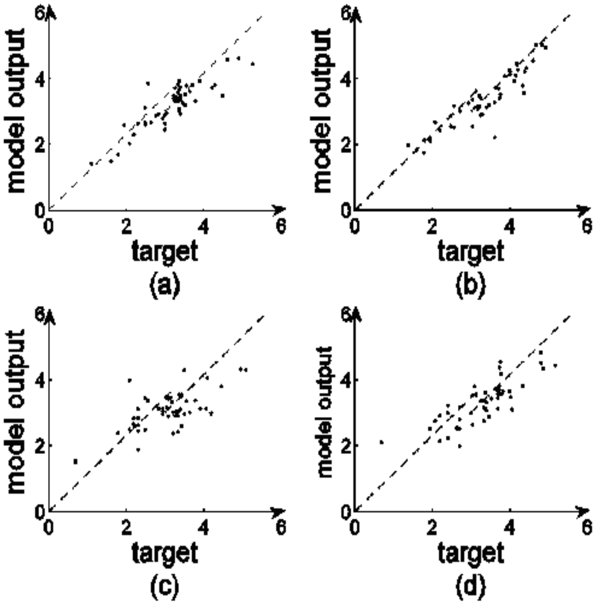


Figure 40. The performance of four lower models (data output versus model output): (a) Q = 0.15 (b) Q = 0.13 (c) Q = 0.14 (d) Q = 0.14

Once the models have been constructed, the prototypes coming from all of them are clustered to form the granular fuzzy model. Following the principle of justifiable granularity, for each of the prototypes obtained here we form the intervals at the output. Several selected results are illustrated in Figures 41-42 for c = 3 and 20.

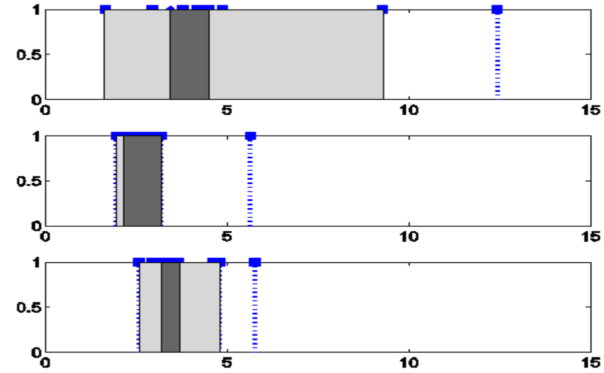


Figure 41. Intervals of prototypes for α equal to 0.01 and 1.6; c = 3

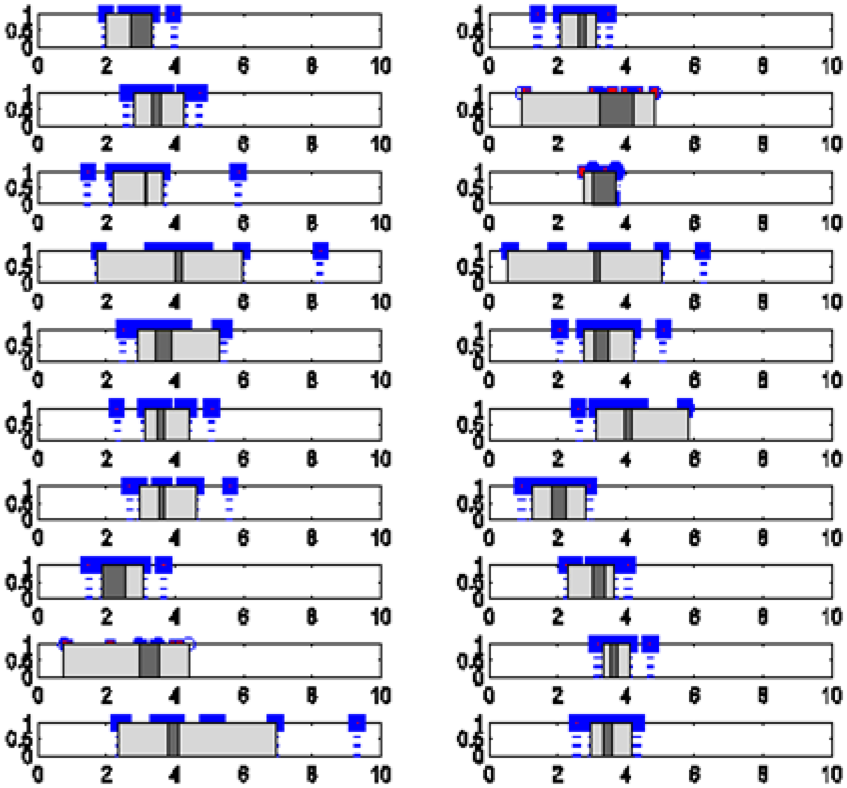


Figure 42. Intervals of prototypes for the values of α equal to 0.01 and 2.96; c = 20

Considering a fixed value of m equal to 2, the corresponding plots of "I" versus the coordinate 6-α are shown in Figure 43. We observe that for higher values of α the values of I are significantly reduced. Some cutoff points can be identified where the values of α are relatively high (so the bounds are quite tight) yet the granular model still captures a significant fraction of data.

The plots in Figure 44 where several prototypes were selected out of all prototypes used by the local models are analogous to those reported in Figure 43.

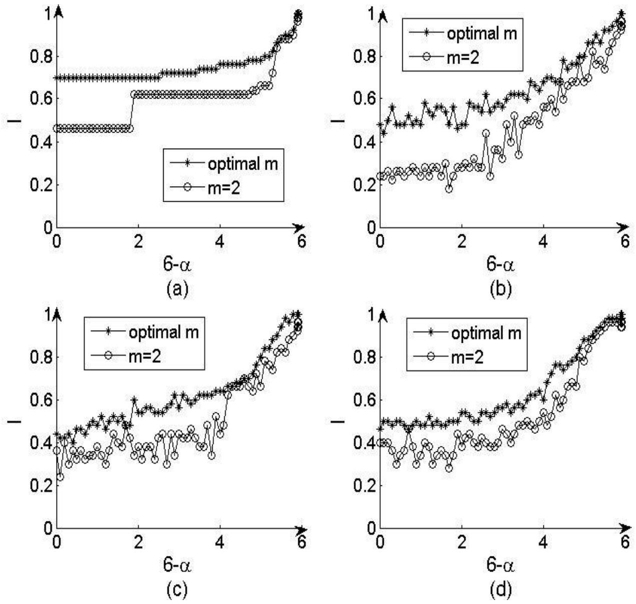


Figure 43. Values of I versus the level of granularity, 6-α for selected values of c: (a) c=3, (b) c=10, (c) c=20, (d) c=40.

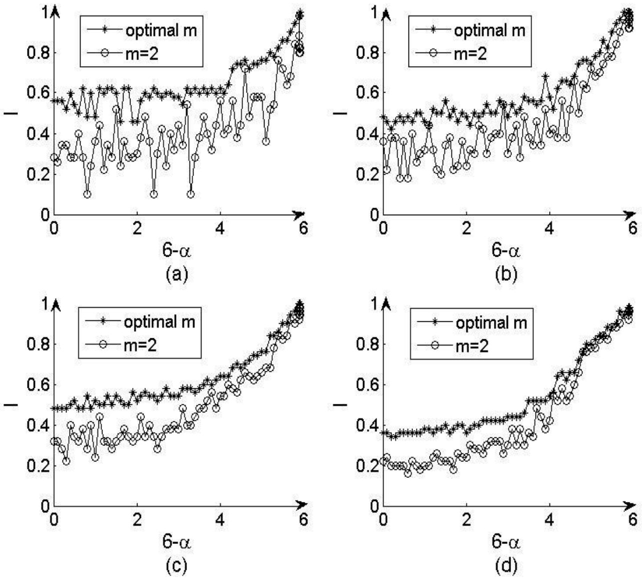


Figure 44. Plots of I versus the level of granularity, 6-α, for selected values of c: (a) c=3, (b) c=10, (c) c=20, (d) c=40.

The values of the AUC obtained for all the models are summarized in Table 13.

Table 13. Values of the AUC for selected numbers of clusters and two strategies of the determination of the prototypes of the granular fuzzy model

Strategy of prototype formation	c = 3		c = 10		c= 20		c = 40	
	Optimal m	m = 2	Optimal m	m = 2	Optimal m	m = 2	Optimal m	m = 2
clustering	4.4040	3.5514	3.7670	2.6678	3.6630	2.9069	3.7247	3.0581
selection	3.7916	2.4319	3.5070	2.6392	3.6470	2.8421	3.1222	2.4868

6.5. Conclusions

The methodological underpinnings of granular fuzzy models stem from the need of an effective conceptual and algorithmic representation and quantification of diversity of locally available sources of knowledge – fuzzy models. The diversity of views at the problem/system/ phenomenon is quantified via the granularity of results produced by the global model constructed at the higher level of a hierarchy. The granular nature of the results formed there is inherent to the diversity of the sources of knowledge. The quantification of granularity itself (viz. the multifaceted nature of available models) is a direct result of multiobjective optimization – it is shown that the criteria of coverage of data and specificity of information are conflicting in nature. The choice of a suitable trade-off in the satisfaction of these two requirements is left to the user. Nevertheless the AUC is helpful here as it can quantify an overall performance of the global model and rank several global models through the use of the AUC values.

Granular fuzzy models subsume the concept of type-2 fuzzy models in the sense they offer compelling, algorithmically well-supported evidence behind the emergence of fuzzy models of higher type. They are more general than type-2 fuzzy models as here we are not confined to any particular architecture of fuzzy models and a way in which type-2 fuzzy sets are incorporated into specific components of the models.

It is worth noting that the notion of granular fuzzy modeling stretches beyond fuzzy models. In essence, as we deal with models articulating locally available knowledge, the quantification of the diversity of such sources becomes encapsulated through information granules produced by granular models. In this way, we can talk about *granular* neural networks (in case of local models being formed as neural networks), *granular* regression (when dealing with individual regression models) or *granular* fuzzy cognitive maps (when the local models are fuzzy cognitive maps), etc. One can look into further generalizations in the form of granular fuzzy models of higher type; say type-2 granular fuzzy models or *granular*^{2} fuzzy models. Those are a result of dealing of several hierarchies of sources of knowledge, namely fuzzy models formed in a two-level hierarchical architecture of the knowledge sources.

7. From Local Neural Networks to Granular Neural Networks

We have elaborated the development of a granular fuzzy model through reconciling and collaboration among local fuzzy models. In this chapter, we devote ourselves to build granular models with another important architecture of system modeling—neural networks. Locally available knowledge in case of neural networks is articulated when being provided with a level of information granularity for output of each neural network. Comparison of development of a granular model with two different fundamental architectures reveals to some extent a general process of system modeling with granular architecture.

7.1. An overall Granular Architectural Blueprint

Neural networks (NNs) can be viewed as essential sources of knowledge. In presence of distributed, individual and locally available data sets generated by the same phenomenon or system being observed from different points of view, we naturally end up with a family of sources of knowledge. Owing to various technical (e.g., available bandwidth) and non-technical limitations (say, privacy) the data cannot be shared. The constructed neural networks can. Forming a *holistic* view at the system by looking at all available sources of knowledge and reconciling them becomes of genuine relevance and practical importance: by doing this we form a general view at the system, perhaps not so detailed and focused as those offered by individual sources of knowledge (neural networks) but providing a holistic and unifying perspective at the problem/system/phenomenon already described locally by the individual neural networks. The diversity of sources of knowledge has to be acknowledged and carefully quantified: this is done by admitting that the results produced by the combined sources of knowledge (networks) have to be non-numeric but rather *granular*.

The concepts of information granules and information granulation have been applied to many fields since the origin of fuzzy sets by Zadeh (Zadeh, Fuzzy sets and information granularity, in: M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.) 1979) (Zadeh, The role of fuzzy logic in the management of uncertainty in expert systems 1983). The information granularity emerging in this construct is a direct manifestation of the diversity of views brought into consideration and as such its constructive role has to be fully acknowledged. The formal realization of information granules could be different including fuzzy sets, rough sets or intervals. The essence of the concept presented above is visualized in Figure 45.

The individual neural networks exposed to any input **x** provide different outputs, which, given their eminent diversity, are then reconciled in the information granulation module. The overall structure emerging in this way is referred to as a granular neural network where the adjective *granular* is directly linked with the nature of the results formed by the neural architecture.

It is worth noting that the local sources of knowledge – neural networks N_1, N_2, .. N_p could be very different in terms of their architectures. The module of forming information granules is central to the overall structure and its design is completed with the use of data to information granule transformation through running a mechanism of justifiable granularity.

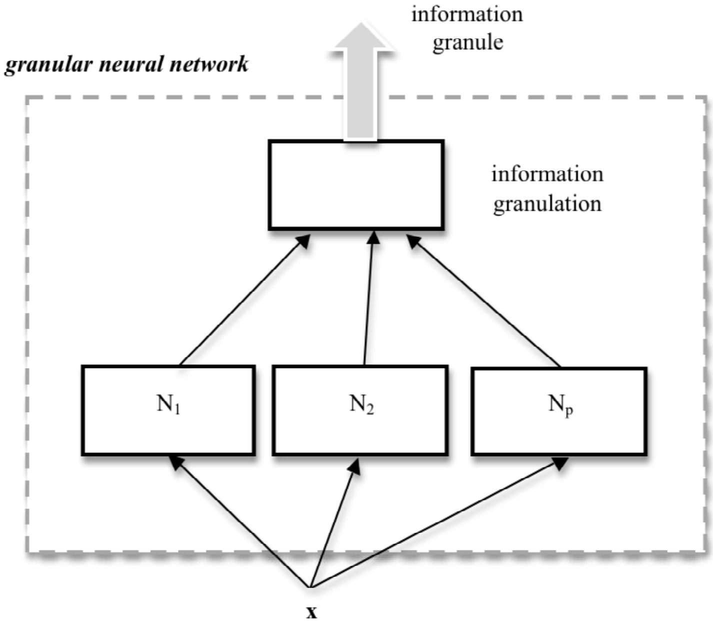


Figure 45. From local neural networks (sources of knowledge) to a holistic view represented in the form of a granular neural network

There have been several studies on the concept of granular neural networks, all of which consider only one neural network with information granules as input and/or output or granular structure. Pedrycz (Pedrycz and Vukovich, Granular neural networks 2001) focused on the formalization of different types of information granules and also provided some architecture (topologies) of granular neural networks. Park *et al*. (Park, Pedrycz and Oh 2009) proposed granular neural networks related to radial based function neural networks while information granules formed in the input and output spaces were constructed by using K-means and Fuzzy C-means, respectively. Zhang *et al*. (Zhang, Jin and Tang, Granular neural networks with evolutionary interval learning 2008) applied evolutionary optimization methods to extract interval-valued granular rules from granular neural networks. Zhang *et al*. (Zhang, Fraser, et al. 2000) made their neural networks process with numerical and linguistic data in one data base and used neural networks to discover granular knowledge from numerical-linguistic data base. Since the emergence of artificial neural networks (Schapire 1990), a new research filed has been opened. We view NNs as an effective tool in approximating any nonlinear problems and apply them to many fields (Pedrycz, Logic-based fuzzy neurocomputing with unineurons 2006) (Nabavi-Kerizi, Abadi and Kabir 2010) (Pedrycz and Aliev, Logic-oriented neural networks for fuzzy neurocomputing 2009) (Du, Li and Fei 2010) (David Sánchez 2002). In the development history, a concept of "neural network ensembles" is worth noting and contrasting with the idea presented here (Akhand, Monirul Islam and Murase 2009) (Das, Turkoglu and Sengur 2009) (Green, et al. 2009) (Kim and Cho 2008) (Granitto, Verdes and Ceccatto 2005). Let us recall the main idea behind the concept of ensemble classifiers (Reformat and Yager, Building ensemble classifiers using belief functions and OWA operators 2008). In essence, in ensemble classifiers one uses a resampling technique in building some local classifiers where each of which is trained on a different training set (Duda, Hart and Stork 2001). Boosting method applies to a set of weak learners and forms a strong learner (Schapire 1990). There are a number of techniques worth recalling here, say, AdaBoost (Freund and Schapire 1997), Gradient Boosting, etc. In all these methods, an entire data set becomes available and it is split randomly. This forms one of the main differences between the boosting techniques and the approach presented in this study. Our primary goal is to design a granular neural network – architecture composed of a number of local neural networks whose results are aggregated in the form of information granules. The granularity of information is an inherent manifestation of the diversity of results provided by sources of knowledge. A way of translating numeric data into meaningful information granules is realized through the principle of justifiable granularity.

7.2. Development Phases of Granular Neural Networks

In comparison with the existing design strategies of neural networks (Pedrycz and Vukovich, Granular neural networks 2001) (Park, Pedrycz and Oh 2009) (Zhang, Jin and Tang, Granular neural networks with evolutionary interval learning 2008) (Zhang, Fraser, et al. 2000) (Pedrycz, Park and Oh, A granular-oriented development of functional radial basis function neural networks 2008) (Panoutsos and Mahfouf 2010), the development of granular neural networks in our study involves two main design phases:

Phase A. Construction of local neural networks. These networks are built separately using individual data sets coming from different sources.

Phase B. Formulation of interval-like information granules. They are formed on basis of numeric outputs developed from the already constructed neural networks.

In the first design phase, a number of local neural networks are trained individually using different data sets. Those neural networks might have different structures and parameters.

The second design phase realizes the essence of the formation of granular architecture combining all local neural networks that have been developed so far. Given some input data, we form the granular neural network by determining the results (outputs) of the individual neural networks and then applying the principle of justifiable granularity. As to the formation of the testing data, two scenarios are envisioned:

(a) The testing data are completely new data, not used in the design of any neural networks formed so far

(b) The testing data are formed by randomly selecting portions of data already used in the training of the local neural networks.

We note that the second scenario is less demanding given the fact that these data were seen by one of the networks in the development; however there is still quite substantial diversity as other data points have not been seen by the granular neural networks.

Depending on whether the quality of individual neural networks is taken into consideration, we either treat all outputs produced by the local networks in the same way or determine the information granules by incorporating the corresponding weights in their design. Here the term "quality" or "performance" is quantified by a certain numeric performance function, such as mean square error (MSE). An easy, yet effective way to assign the values of the weights is to take into account the performance of local networks. The way to compute the weights is to calculate the following ratios

$$\beta_i = 1 - \frac{Q^i_{test}}{Q^i_{train}} \tag{7.1}$$

where Q_{test} and Q_{train} are the values of the performance index determined by the training and testing data used in the construction of local neural networks. The weights are obtained by normalizing the values of β_i.

In summary, the overall design process consists of the following steps:

Step 1. Arrange "g" local data sets for the individual neural networks and one testing data set (denote it by TE). The arrangement of the data was discussed in the previous section.

Step 2. Construct local neural models. The offline backpropagation algorithm (BP) is considered as a sound learning option. All the networks have the same topology--multiple layer perceptrons (MLP). The parameters are optimized individually by considering the given "g" data sets. The resulting neural networks are denoted by NN_1, NN_2, …, NN_g.

Step 3. Assess the quality of NN_1, NN_2, …, NN_g following (7.1).

Step 4. Construct information granules for given input data by running the principle of justifiable granularity.

Step 5. Evaluate the performance of granular neural networks. Here we are interested in determining how well the information granules “cover” the data.

Given a certain value of α (the parameter used in the construction of the corresponding information granules, see chapters 3.2 and 6.2), we compute a fraction of output data included in the intervals produced by the information granules, please refer to (3.15). The granular network with higher value of AUC is preferred (see chapter 3.3).

7.3. Experiments

In the collection of the experiments reported in this section, we use three data sets namely abalone, auto MPG and Boston housing. As local neural networks, we consider multiple layer perceptrons (MLPs) with a single hidden layer constructed by running the backpropagation learning and the neurons equipped with a sigmoidal transfer function. The neurons in the output layer come with a linear transfer function. Before we resort to neural networks, the raw data are normalized. There are also some parameters of the algorithm, which can affect the performance of our granular neural networks. To analyze their influence, we illustrate the performance when using various values of the parameters.

Abalone dataset

This data set consists of 4,177 data points and each data point is described in the 8- dimensional feature (input) space. The output is the age of abalone. See http://archive.ics.uci.edu/ml/datasets/Abalone.

We randomly select 95% of the original data to build local networks. Whereas the testing data can either be composed of the other 5% of the original data (scenario (a)) or be chosen from the 95% of original data (scenario (b)). There are 10 local networks in this example (in each 80% of local data are used for training networks and 20% are used for testing) and the number of local networks depends on how many distributed individual data sets we use.

The selected individual network's output and its associated real target are illustrated in Figure 46.

In the sequel, Figure 47 shows all local neural networks' performance expressed by the values of the performance index obtained on the training and testing set. More specifically, we display the values of Q_{test} versus Q_{train}. In most cases, as could have been expected, $Q_{test} > Q_{train}$ while the ratio Q_{test}/Q_{train} is around one.

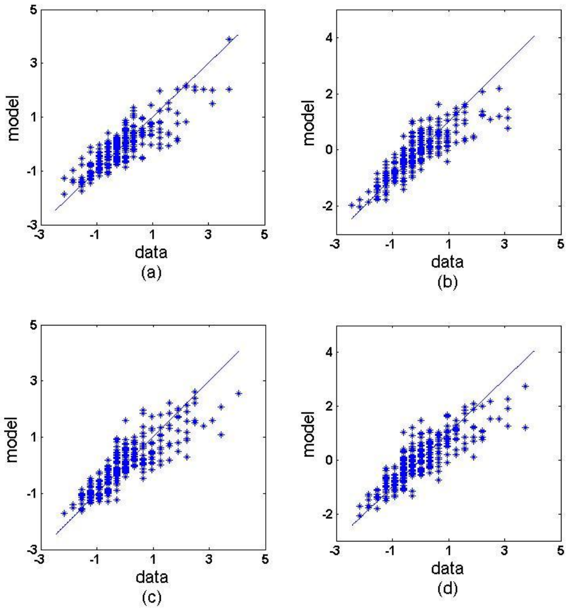


Figure 46. Performance of four selected local models: data versus model output

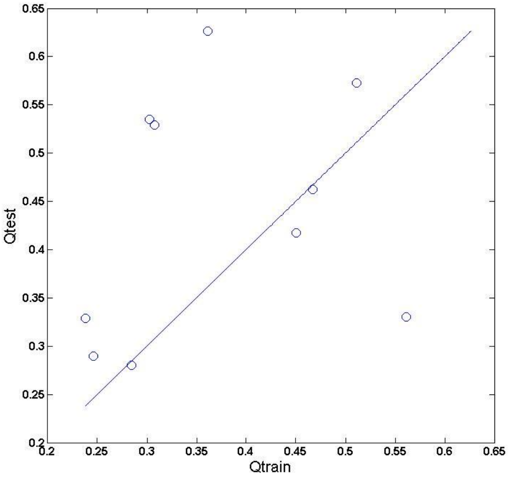


Figure 47. Q_{train} versus Q_{test} for each local network

A series of intervals are formed in the second phase of the design of a granular neural network. Figure 48 shows the lower and upper bounds of the output of testing data (formed by scenario (a)) and the circles scattering on the plots are real outputs. For this data set, the value of α does not exhibit a significant influence on the coverage rate.

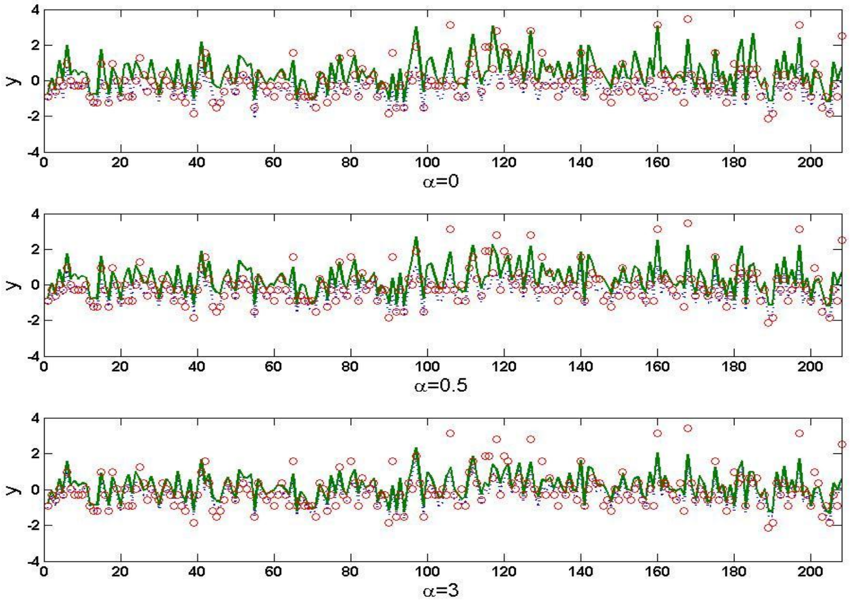


Figure 48. Lower and upper bounds of the output of granular neural network versus the data, results reported for selected values of α; α = 0, 0.5 and 3.0

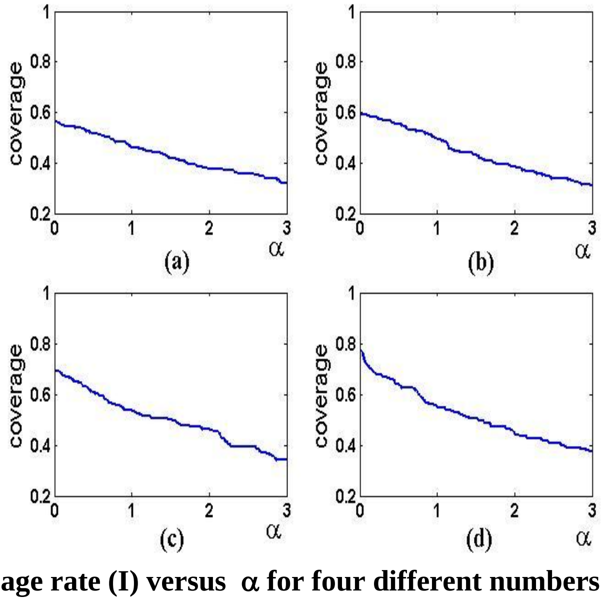
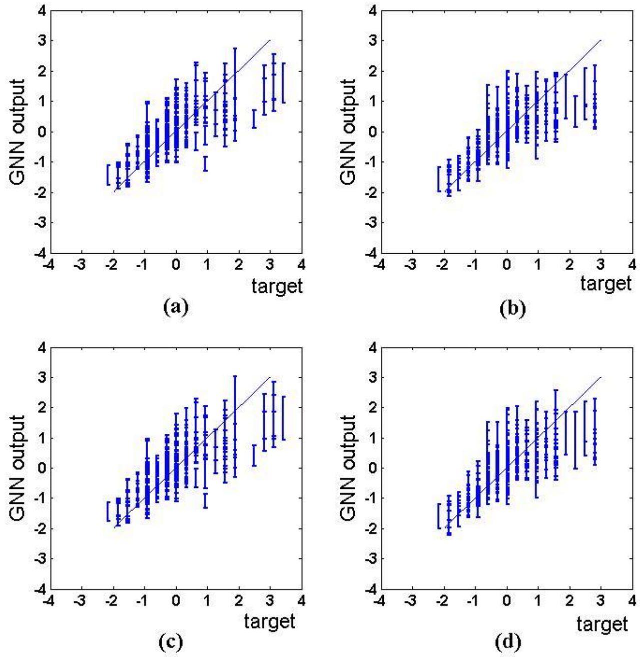


Figure 49. Coverage rate (I) versus α for four different numbers of local models: 8, 20, 30, 40. The corresponding values of the AUC are equal to 1.29, 1.32, 1.49, and 1.55, respectively



(a) (b)

(c) (d)

Figure 50. The output of the GNN versus target in case of 10 local networks: (a) weights are equal to one and a new testing data; (b) all weights are equal to one and the testing data selected from the training data (c) weights are not equal and testing data are completely new (d) weights are not equal and testing data are selected from training data

An impact of the values of α on the coverage rate is illustrated in Figure 49. A monotonic relationship is noticeable: higher values of α result in narrow intervals and this translates into lower coverage values. One can observe that the reduction in the coverage is reduced at a similar rate. The values of AUC offer a global view at the behaviour of the granular model with the highest value reported in case of dealing with 40 local networks.

Figure 50 displays the outputs of granular neural network versus the target values. Most target values are covered by the intervals for the four different considerations. Also we find that this algorithm has weaker ability on larger values of target.

Auto MPG dataset

The details of this data set have been described in chapter 5.5. The organization of the training and testing data as well as a split of the training data are the same as in the case of the previous data set. The performance of four selected local neural networks is illustrated in Figure 51.

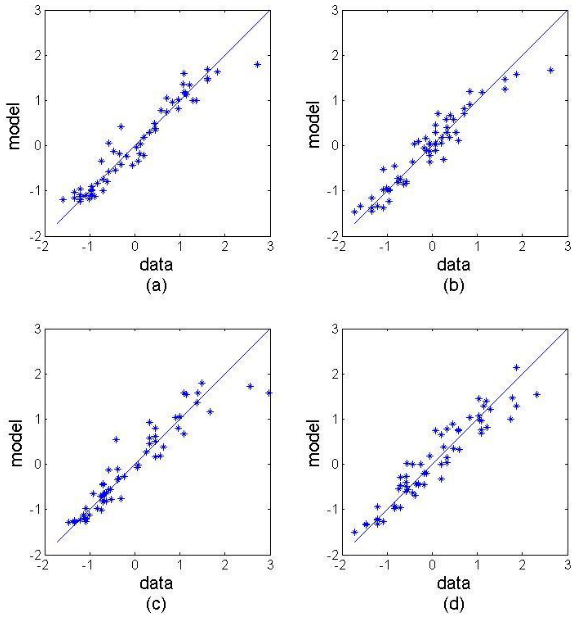


Figure 51. Performance of the local models: data versus model output

The bounds of intervals produced by granular neural network are shown in Figure 52. As before, the plots are shown for selected values of α; it is worth noting that no substantial changes are reported for values of α higher than 3.0.

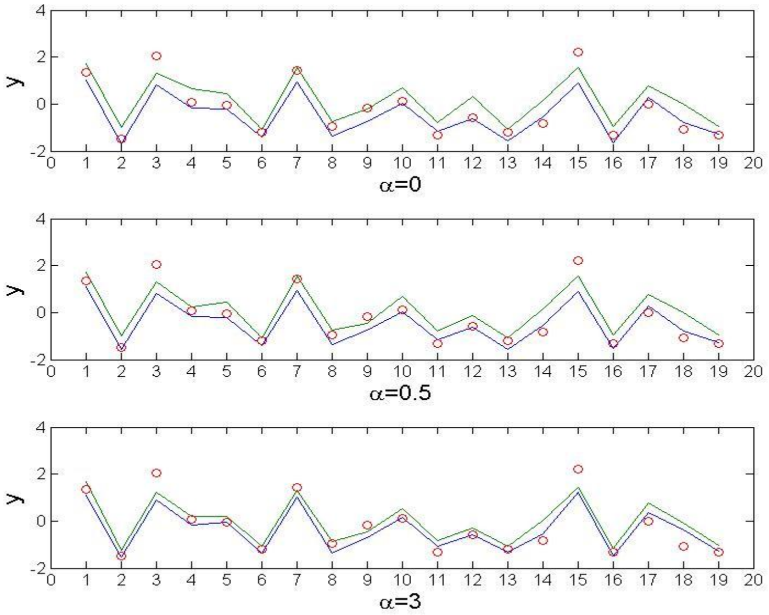


Figure 52. Lower and upper bounds of the output of granular neural network versus the data, results reported for selected values of α: α = 0, 0.5 and 3.0

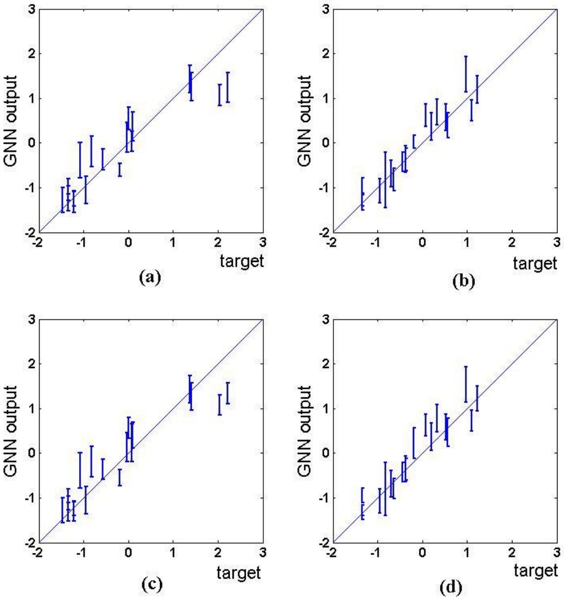


Figure 53. The GNN output versus target when the number of local models is 5. (a) weights are equal to one and testing data are completely new (b) weights are equal to one and testing data are selected from training data (c) weights are not equal and testing data are completely new (d) weights are not equal and testing data are selected from training data

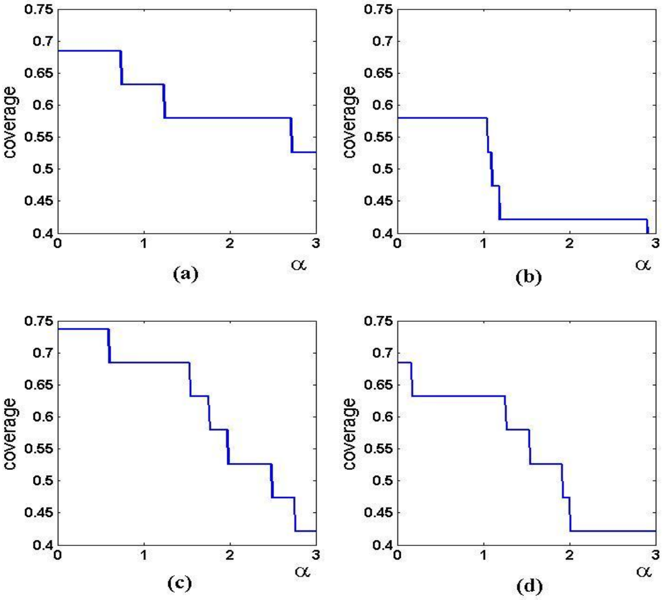


Figure 54. Coverage rate (I) versus α for four different numbers of local models: 4, 5, 6, 7, the AUC are: 1.83, 1.43, 1.85, 1.63

In Figure 53, we present granular output (interval) of networks and real data. In case that testing data were selected from the training data: (b) and (d); the resulting intervals of the output cover more data that in the two other cases (a) and (c).

The coverage rate (when α is equal to zero) and the value of AUC do not always increase when the number of local models increases from 4 to 7, see Figure 54. The highest coverage rate in these four cases reaches 0.76.

Boston Housing dataset

The Boston Housing data set, comprising 506 instances, 13 input variables (features) and a single continuous output, concerns real house price estate in the Boston area. The organization of the training and testing data is the same as before. The training data set is split into 7 subsets (each comprises 80% for training networks and 20% testing data) using which local neural networks are constructed.

Both the coverage rate and AUC are largest comparing to the other two data sets, see Figure 55. The largest coverage is around 0.85 when the number of local models is 6 and the corresponding AUC is 1.99. However, the largest AUC (2.07) occurs when the number of local models is 4.

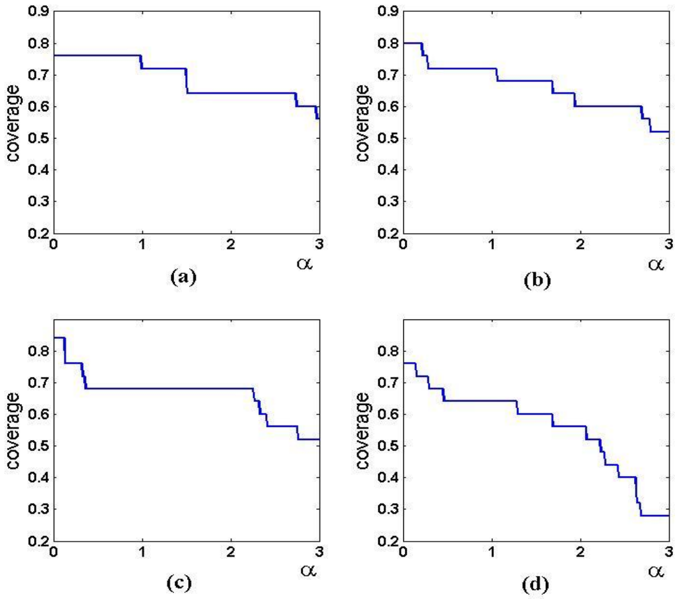
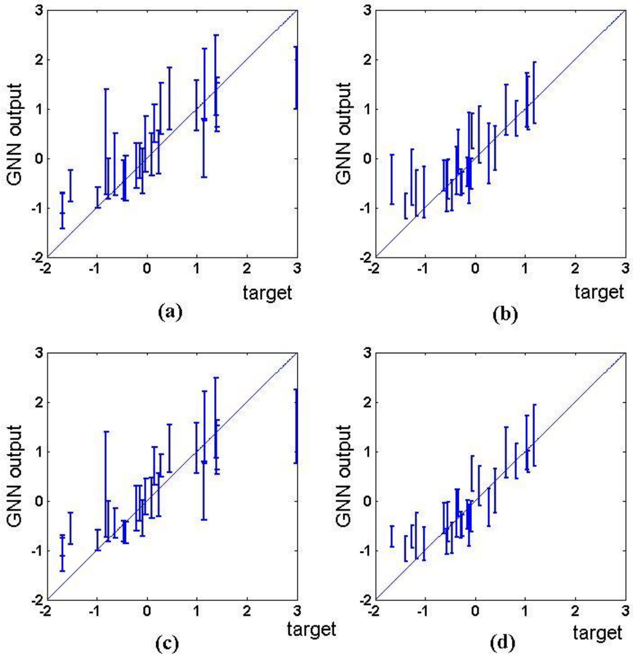


Figure 55. Coverage rate versus values of α on four different numbers of local models: 4, 5, 6, 7, the AUC are: 2.07, 1.99, 1.99, 1.67



(a) (b) (c) (d)

Figure 56. The GNN output versus target when the number of local models is 7. (a) weights are equal to one and testing data are completely new (b) weights are equal to one and testing data are selected from training data (c) weights are not equal and testing data are completely new (d) weights are not equal and testing data are selected from training data

In light of Figure 56, we conclude that the granular neural network exhibits better performance when the testing data are selected from the training data used for local neural networks as opposed to those composed of data never used before in the design process.

7.4. Conclusions

In this study, we have introduced a concept of granular neural networks being as a result of forming a holistic view at individual sources of knowledge represented by local neural networks. We have emphasized the need to capture and quantify a diversity of the sources of knowledge being dealt with, which has resulted in a granular nature of the aggregation mechanism and introduced a novel architecture of granular neural networks. While we have advocated the use of information granularity in capturing and quantifying the diversity of views, one can investigate the concept of granularity in terms of information needs and cognitive capabilities or skills. The idea arises when one considers to insert information granules into neurocomputing and to explore how neural networks work on partial dataset. The experimental results reveal some interesting and practically viable findings:

- The adjustable interval information granules controlled by one or more parameters make the information granules meaningful and changeable. Since the principle is realized on some predicted outputs, these intervals can cover the real outputs with a comparable high confidence also with rational meaning themselves.
- An interval of expected output is obtained with our algorithm by firstly applying data to some learned local models and then using the principle of justifiable granularity. This two-step method successfully solves the limitation of spatial and temporal sources of data and low prediction accuracy.
- Knowledge or structures learned from either neural networks or adjustable information granules helps reveal relationship between data. The proper modeling methods and reasonable intervals are essential.

8. Local Granular Neural Networks

The previous chapter introduces a way to form granular neural networks by individual available networks in which information granularity is used to quantify the diversity of sources. In this chapter, we propose another method of building a granular neural network based on a single local network which implies the development process: first train a neural network with some method and then build information granules round the numeric connections of the network and form a granular architecture. The allocation of information granularity is used here to achieve an optimal architecture of the granular network.

8.1. The Concept of Local Granular Neural Networks

Most neural networks encountered in the literature are *numeric* constructs realizing a certain nonlinear mapping. A conceptually viable and practically useful generalization of numeric neural networks comes in the form of a non-numeric mapping realized by neural networks. In this case, we may refer to such networks as *granular* neural networks. The non-numeric (granular) nature of the mapping arises because of the granular character of the connections. In this case any numeric input to such network produces a granular output of the network. There are several compelling reasons behind the realization of this type of neural networks. First, by establishing granular outputs one can effectively gauge the performance of the already constructed numeric neural network in presence of training data. Second, when dealing with new data, the network forms granular outputs, which are instrumental in the quantification of the quality of the obtained result. For instance, in case of prediction we are provided with a comprehensive forecasting outcome; instead of a single numeric result, formed is an information granule whose location and size lead are highly descriptive of the quality of prediction.

The term “granular” pertains to the nature of the developed construct and by no means is confined to a certain specific type of the neural network. Instead it concerns a general augmented neural architecture it builds upon. The proposed concept applies equally well to multilayer perceptrons (MLPs) or radial basis function neural networks (RBFNN)) cf. (Ishibuchi and Tanaka, An architecture of neural networks with interval weights and its application to fuzzy regression analysis 1993) (Ishibuchi, Kwon and Tanaka, A learning algorithm of fuzzy neural networks with triangular fuzzy weights 1995) (Kuo, Wu and Wang 2002) (Huang, Zhang and Huang 1998) (Park, Pedrycz and Oh 2009) (Zhang, Jin and Tang, Granular neural networks with evolutionary interval learning 2008) (de Weerdt, Chu and Mulder 2009) (Garczarczyk 2000) (Pedrycz and Vukovich, Granular neural networks 2001). It works well with neurofuzzy systems cf. (Bortolan 1998) (Buckley and Hayashi 1994) (Liu and Li 2004) (Juang and Lin, A recurrent self-organizing neural fuzzy inference network 1999) (Y. Ishibuchi 1996)

(Juang, Huang and Lin, A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing 2009) (Melin, Mendoza and Castillo 2011) (Wang, Cheng and Lee 2004) (Juang, Huang and Cheng, An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems 2010).

At this point, it is instructive to relate the proposed approach with what is known in the literature as interval neural networks, especially interval MLPs, see (Ishibuchi and Tanaka, An architecture of neural networks with interval weights and its application to fuzzy regression analysis 1993) (de Weerdt, Chu and Mulder 2009) (Ishibuchi and Nii, Improving the generalization ability of neural networks by interval arithmetic 1998) (Fang, et al. 2006) (Garczarczyk 2000). Ishibuchi *et al* in (Ishibuchi and Tanaka, An architecture of neural networks with interval weights and its application to fuzzy regression analysis 1993) proposed a MLP neural network with interval weights and interval biases and derived a learning algorithm supporting its development. The studies presented in (Ishibuchi, Kwon and Tanaka, A learning algorithm of fuzzy neural networks with triangular fuzzy weights 1995) being a continuation of the work reported in (Ishibuchi and Tanaka, An architecture of neural networks with interval weights and its application to fuzzy regression analysis 1993) generalize numeric inputs of fuzzy neural networks to their fuzzy set-based counterparts. In (Zhang, Fraser, et al. 2000), a granular neural network using backpropagation algorithm and fuzzy neural networks is used to handle numeric-linguistic data fusion providing a mechanism of knowledge discovery in numeric-linguistic databases. It is noticeable that in all these cases, the corresponding interval neural networks are built from scratch.

In contrast, in the proposed neural architecture, we are concerned with granular connections, in general, and this implies various formal models of information granules including sets, fuzzy sets, probabilities, rough sets. From a computational perspective, intervals offer a computationally appealing alternative and while the underlying concepts are of general character and those are relevant to other formalisms of information granules, more detailed investigations will be focused on interval-valued (interval) neural networks. Furthermore given the focus of the study, the term information granule and interval are used interchangeably. The entire design process behind the proposed networks is outlined as follows. A starting point of the overall design process is a numeric network that has already been developed by means of one of well-established learning strategies. Then a data set (the same as training data set or a new one) is used to construct a granular network, viz. form interval connections on a basis of the given network. In this sense, the resulting granular construct augments the topology of the existing neural architecture. The design process (viz. the formation of information granules of the connections) is well articulated and translates into an optimization problem. Information granularity is regarded as an essential design asset and an allocation of granularity following some protocols leads to the optimization of some performance index gauging the quality of the resulting granular neural network.

Along with the granular (interval) neural networks, some comments are worth making in the context of applications of such constructs. The granular network produces results being in better rapport with reality when dealing with training data. It is accomplished by showing intervals of possible output values rather than a single numeric entity. An interesting application scenario refers to knowledge (experience) transfer as elaborated on in (Pedrycz, Russo and Succi, Knowledge transfer in system modeling and its realization through an optimal allocation of information granularity 2012). A neural network is built on a basis of existing data. Then one intends to use it for a new yet somewhat related problem. The currently available data are very scarce so building a sound model on its basis is not justifiable. In this case, we consider the usage of the existing neural network, viewing it as a source of knowledge. However being cognizant of the differences (and some similarities) of the current circumstances and the environment within which the neural network was designed, one can anticipate that the results produced by the network have to be treated with caution and represented as information granules instead of single numeric entities. This is accomplished by forming a granular neural network on a basis of the original network that is made granular (accepting interval connections) on a basis of the existing numeric evidence (experimental data).

The main objective of this study is to develop a comprehensive design methodology for granular neural networks through an optimal allocation of information granularity where the levels of information granularity distributed throughout the network are assigned to each connection in an optimal fashion. In our study, we offer a sound motivation behind the emergence of granular connections. Information granularity is regarded as an important and practically useful design asset whose optimal allocation helps optimize the resulting construct. It is also worth stressing that granular neural networks are constructed based on the already developed numeric neural network (we are not concerned about a way in which the design has been completed). In essence, the design of the granular neural network comes as an enhancement of the well-established practice of the formation of neural networks and augments the originally formed network with a sound quantification of features of the numeric construct.

The concept of information granules is formalized in different ways depending upon the underlying formalism. Fuzzy sets, sets, rough sets are just representative examples of information granules. The number of information granules implies a level of generality one assumes when dealing with a problem at hand. The level of abstraction (generality) associates with information granularity. In a nutshell, information granularity is concerned with a “size” (specificity) of information granules. Depending upon the formalism being used, information granularity can be described and quantified in different ways. Commonly, as we are focused on a size of the granule, its granularity is expressed by counting the number of elements embraced by the granule (discrete case) or the length of the granule (continuous case). In situations where there are grades of membership involved (like in fuzzy sets), certain generalizations are sought such as a sigma-count of fuzzy sets (where a summation is completed over the grades of membership) is involved.

8.2. A Design of Interval-Valued Neural Networks

The architecture of a granular neural network along with the associated learning scheme are formed by starting with a certain already designed (numeric) neural network and augmenting its structure by considering granular connections spanned over the numeric weights (connections).

Here we consider a MLP with a single hidden layer. From numeric neural network to a granular neural network, the design process is discussed below:

As indicated, in our design, we proceed with the already constructed MLP (e.g., realized in terms of batch backpropagation (BP) learning method, see (Ridella, Rovetta and Zunino 1997)). The choice of the size of the hidden layer is decided upon during this design phase. The activation functions of the neurons in hidden layer and output layer are denoted by f_1 and f_2, respectively. The detailed formulas are outlined as follows:

The original network:

hidden layer: $o_j = f_1(z_j)$, $z_j = \sum_{i=1}^{n} w_{ji} o_{ti} + \theta_j$, j = 1, 2, ..., n_1, $f_1(x) = \frac{2}{1+e^{-2\times x}} - 1$

output layer: $y = f_2(z)$, $z = \sum_{j=1}^{n_1} w_j o_{tj} + \theta_j$, $f_2(x) = x$

When it comes to the interval-valued neural network, the above formulas are generalized in the following way

hidden layer: $o_j = [o_j^{-}, o_j^{+}] = [f_1(z_j^{-}), f_1(z_j^{+})]$

$z_j^{-} = \sum_{i=1}^{n} (w_{ji}^{-} o_i + \theta_j)$, $z_j^{+} = \sum_{i=1}^{n} (w_{ji}^{+} o_i + \theta_j)$, j = 1, 2, ..., n_1,

output layer: $Y = [y^{-}, y^{+}] = [f_2(z^{-}), f_2(z^{+})]$,

$$z^{-} = \sum_{i=1}^{n} (\min(w_j^{-} o_j^{-}, w_j^{+} o_j^{-}, w_j^{-} o_j^{+}, w_j^{+} o_j^{+}) + \theta_j)$$

$$z^{+} = \sum_{j=1}^{n_1} (\max(w_j^{-} o_j^{-}, w_j^{+} o_j^{-}, w_j^{-} o_j^{+}, w_j^{+} o_j^{+}) + \theta_j)$$

As the outputs of the granular neural network are intervals while the targets coming from the experimental data are numeric, we have to define a suitable performance index (objective function), whose optimization (maximization or minimization) is realized through a suitable allocation (distribution) of information granularity.

The algebraic operations of intervals are introduced in chapter 2.1. The random allocation of information granularity is not a difficult thing; however, how to make the constructed granular network optimal in terms of a certain performance index is challenging. Hence we define a fitness function which comprises two criteria people always interested in. An optimization mechanism is as well used to execute the construction.

As we are concerned with the granular output of the network, which has to be evaluated with regard to the numeric target, two criteria (performance indexes) are worth considering. The first one looks at the quantification of the concept of coverage – an extent to which the target values are “covered” (included) in the corresponding granular outputs. Another one is focused at expressing a level of specificity of the information granules produced by the network.

Let us assume that for the purpose of the evaluation, we have some data ***F*** coming in the form the pairs ($\mathbf{x}_t$, target$_t$), t=1, 2, …, N.

Coverage The coverage criterion is quantified by some index Q, which is described as a ratio

$$Q = \text{number of data inside the intervals formed by the granular neural networks}/N \tag{8.1}$$

Specificity is expressed in terms of the average length of the intervals produced by the granular neural network for the inputs coming from the data set ***F***.

$$\text{length} = \frac{\sum_{t=1}^{N} \left| y_t^{+} - y_t^{-} \right|}{N} \tag{8.2}$$

In the optimization of the allocation of granularity, we consider several design scenarios:

(i) use of the coverage criterion. This gives rise to a single-objective optimization.

(ii) use of both criteria, coverage and specificity. These two criteria, given their character, are very likely in conflict. In this case a two-objective optimization producing a Pareto front is an option worth pursuing.

In this section, we briefly review the two essential modes of PSO, namely its single-objective and multiple objective (two-objective) versions. These two are used in the subsequent experiments reported in this study.

In PSO, a swarm of particles where each of them represent a potential solution (in our case it is a vector of optimized levels of information granularity). In this study, the particles represent possible solutions of the levels of granularity: ε_1, ε_2, …, ε_h, for protocol C_3 and ε_{1-}, ε_{2-}, …, ε_{h-}, ε_{1+}, ε_{2+}, …, ε_{h+} for protocol C_4. In other words, the searching space is $[0, \varepsilon]^h$ for protocol C_3, and $[0, \varepsilon]^{2\times h}$ for protocol C_4. h is the total number of connections and biases of the network. We initialize the particles randomly to proceed with the search of the space. For protocol C_3, D = h; for protocol C_3, D = 2*h. The pseudo codes of single objective PSO are list in chapter 2.4.

PSO is also of interest in the context of multiobjective optimization considering its rapid convergence and a relatively low computing overhead, (Coello Coello, Pulido and Lechuga, Handling multiple objectives with particle swarm optimization 2004) (dos Santos Coelho, Barbosa and Lebensztajn 2010) (Fonseca and Fleming 1998). Typically, we consider two objectives, which as in the study here, are in conflict. The results give rise to a Pareto front forming a collection of non-dominated vectors. The basic concepts of Pareto optimality, Pareto dominance and Pareto front are introduced in (Coello Coello, Pulido and Lechuga, Handling multiple objectives with particle swarm optimization 2004). Pareto optimal solutions are also termed noninferior, and their corresponding vectors are termed non-dominated vectors cf. (Horn 1997). To construct the front, we use the approach presented in (Coello Coello, Pulido and Lechuga, Handling multiple objectives with particle swarm optimization 2004).

At each generation, we calculate the values of the two objectives for each particle of the swarm, namely its coverage and specificity. The values of specificity and the complement of the coverage, that is 1- coverage, form a two-dimensional vector. Then we find non-dominated vectors in the entire population and compare them with regard to the external repository. At the beginning of the process, this repository is empty. If the external repository is empty, then the current solution is accepted and stored there. Otherwise, we compare the new solution with the existing non-dominated vectors present in the external repository. If this new solution is dominated by an individual within the external archive, then such solution is discarded. If none of the elements in the external archive dominates the new solution, then such solution is stored there. If there are solutions in the archive that are dominated by the new solution, then such solutions are removed from the archive. Finally, the elements of the external repository are used to construct the Pareto front. To visualize the Pareto front, we display all solutions present in the external archive in a two-dimensional space with specificity-(1-coverage) coordinates.

To facilitate a quantitative assessment of the performance of the two-objective optimization algorithm and single-objective optimization algorithm, we compute an average distance (Euclidean one) between points positioned on the Pareto front and the origin of the coordinates. The pseudo codes of single objective PSO have been given in chapter 2.4.

8.3. Experimental Studies

In this section, we present a series of numeric experiments to illustrate the proposed algorithm, show its development and quantify the resulting performance.

The data sets used here are: synthetic data, auto MPG, Boston Housing, PM10 and Bodyfat. In all the experiments, we randomly split the data into the training (80%) and the testing (20%) set. The performance of the numeric neural network is quantified by means of the commonly used Mean Squared Error (MSE).

In all experiments, we start with a neural network with a single hidden layer. The activation function used for the neurons in the hidden layer is sigmoidal while for the neurons in the output layer we use a linear function. The number of neurons in the hidden layer was selected experimentally by analyzing the performance of the network vis-à-vis the size of the hidden layer. The training of the network was realized using a standard Levenberg-Marquardt minimization method, which was run for a maximum of 10,000 epochs (this number was sufficient given a fast rate of convergence of the method). The data were pre-processed by normalizing the input variables to the unit interval. The initial values of the weights (connections) were set up randomly to be within the range [-1, 1].

The construction of the granular neural network was realized by running the protocols of information granularity allocation as presented before and using the training data (the same as being utilized in the formation of the original neural network) – case (a) and the testing data – case (b). The setup of the PSO is as follows: number of generations - 100, the size of population- 50. The values of these parameters are in line with those reported in the literature (Kennedy and Eberhart, Swarm Intelligence 2001). We also ran some other combinations of the size of the population and the number of generations but no improvement of the fitness function was reported when increasing the values of these two parameters. Computing overhead is another major concern when we compare different protocols of granularity allocation. To quantify it, we report the actual running time for protocols C_1 - C_4.

To quantify the performance of the granular neural network at the global level (irrespectively of a specific value of ε), we compute the area under curve (AUC) of coverage relationship.

Two input synthetic data

The two-variable function is a sine wave, which has been used as regression benchmark data in (Wedge, et al. 2006).

$$y = 0.8 \times \sin(x_1/4) \times \sin(x_2/2) \qquad (8.3)$$

where x_1 is in [0, 10] and x_2 is in [-5, 5]. The training data are made up of 480 randomly selected input-output pairs (followed a uniform distribution over the input space). The function along with the superimposed training data is illustrated in Figure 57.

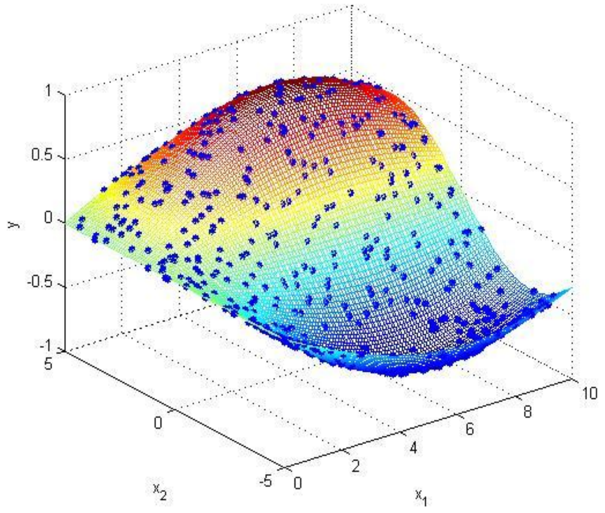


Figure 57. Nonlinear two-variable function along with training data

The performance of the constructed neural network quantified in terms of the MSE is illustrated in Figure 58. Through a visual inspection, we choose the number of neurons in the hidden layer to be equal to 8.

Let us start with the optimization with the fitness function being treated as the coverage and the data formed following case (a). The plots of the coverage regarded as a function of ε are included in Figure 59. The values of the AUC are computed over the unit interval however the plot is shown only for a portion of the entire range of ε.

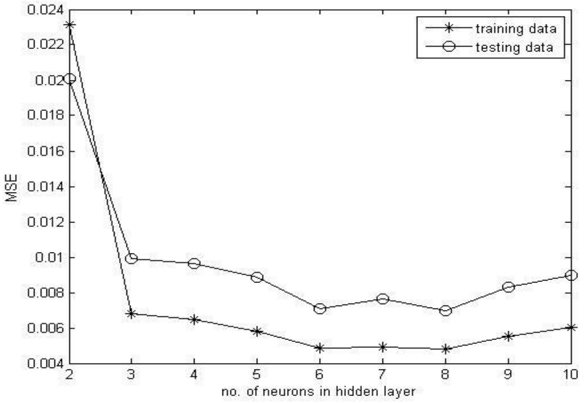


Figure 58. Performance index as a function of the number of neurons in the hidden layer

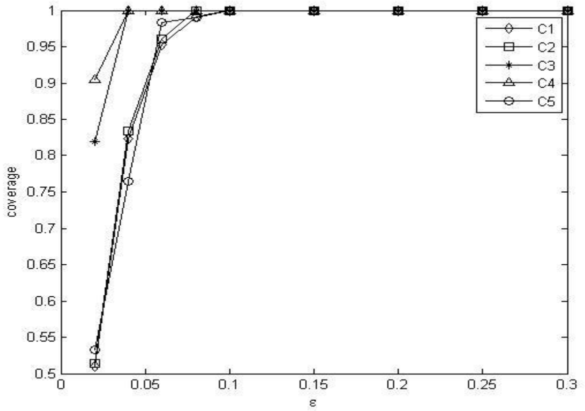


Figure 59. Coverage as a function of ε for the protocols of information granularity allocation used in the study

The corresponding values of the AUC on training data are: protocol C_1: AUC = 0.9755, protocol C_2: AUC = 0.9762, protocol C_3: AUC = 0.9864, protocol C_4: AUC = 0.9881 protocol C_5: AUC = 0.9754.

As shown in this figure, the value of AUC obtained for protocol C_4 is slightly higher than the one produced for C_3. However, both are higher than the ones produced by other three protocols. This is not surprising as they reflect the increasing flexibility of the successive protocols of allocation of granularity and a better, more effective usage of information granularity.

Figure 60 displays the values of the fitness function reported in consecutive generations; it is noticeable that the convergence of the optimization process (increasing coverage) occurs in the first 25 generations.

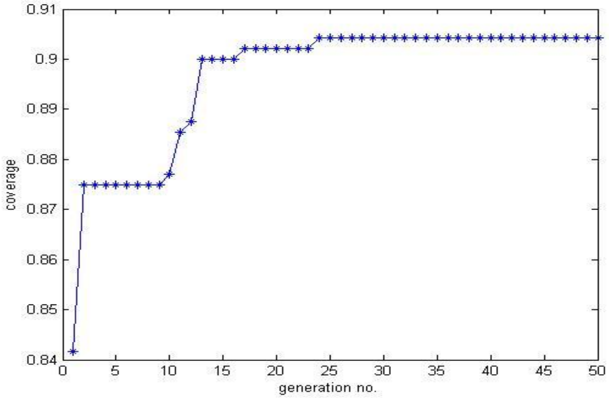


Figure 60. Performance of PSO expressed in terms of the fitness function obtained in consecutive generations; ε = 0.02, protocol C_4

The granular connections of the network are displayed in Figure 61 (ε = 0.02) and Figure 62 (ε = 0.2); both the lower and upper bounds of the intervals of the connections are visualized.

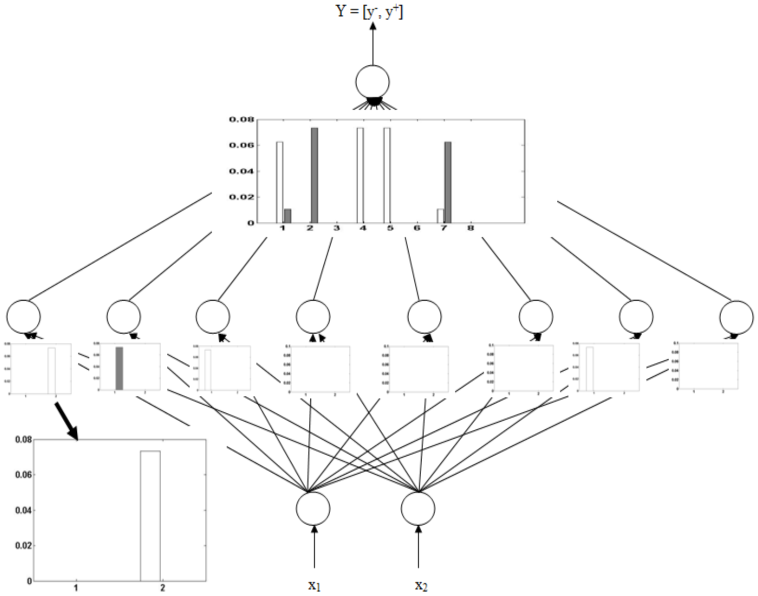


Figure 61. The optimized allocation of granularity levels realized by running protocol C_4 with ε = 0.02. The biases of the neurons are not shown.

The optimized coverage of this case is 0.9042. We use the same range for y axis on each plot of intervals of granularities. Refer to the enlarged plot of the first hidden neuron's for more detail. Since each interval is constructed by extend the numeric value to a lower bound and an upper bound separately, we actually form two sub-intervals for each connection. And in this way, we have to distribute the bounds of granularity to each connection. There are some connections which have zero values of the levels of granularity, which means that these connections are effectively retained as numeric ones.

When considering case (a), the obtained granular neural network is displayed in Figure 61. As to the case (b) of ε = 0.02 it is obvious that there are many connections remaining numeric instead of intervals in the optimized vector of ε.

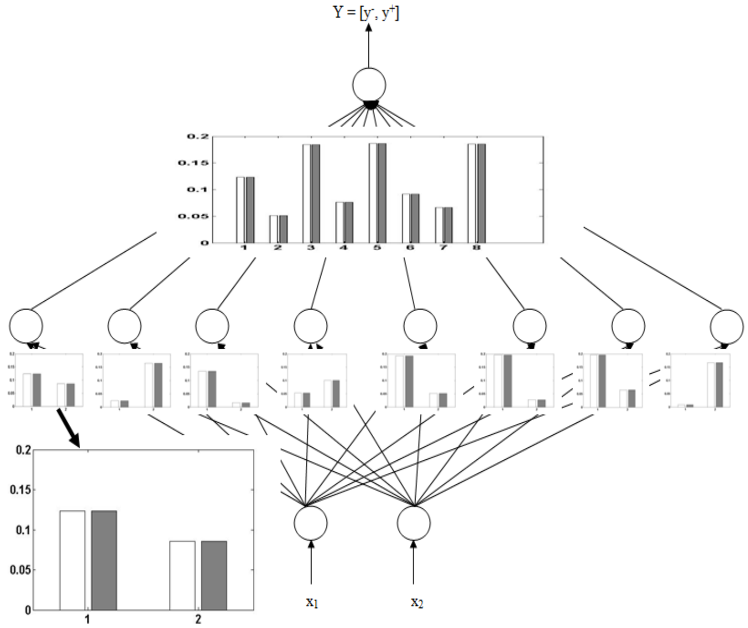
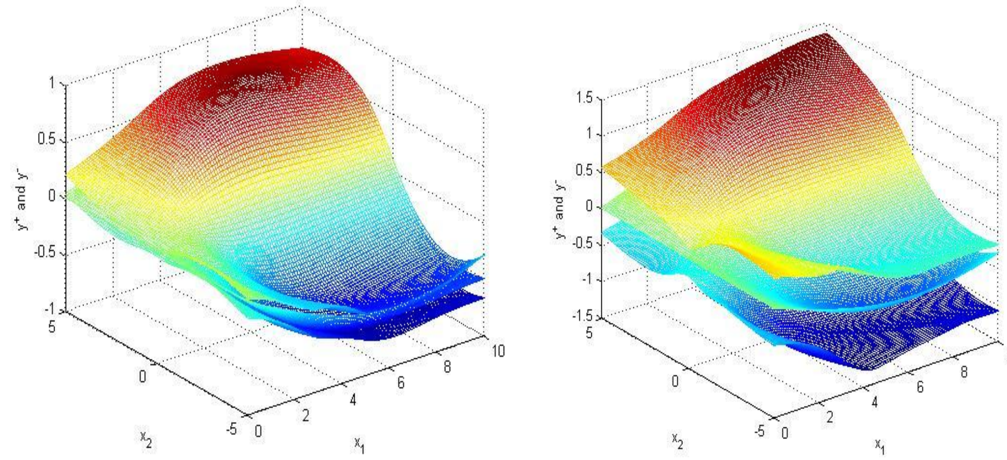


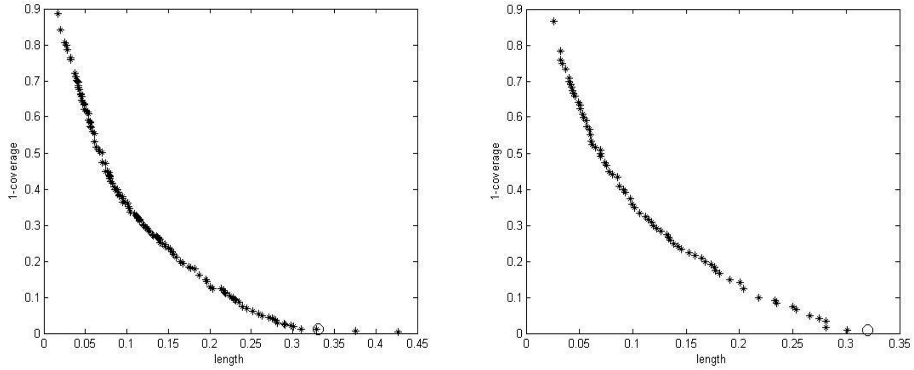
Figure 62. The optimized allocation of granularity of level 0.2 (ε) for C_4



(a) (b)

Figure 63. Outputs of the granular network of case ε = 0.02 (a) and ε = 0.2 (b)

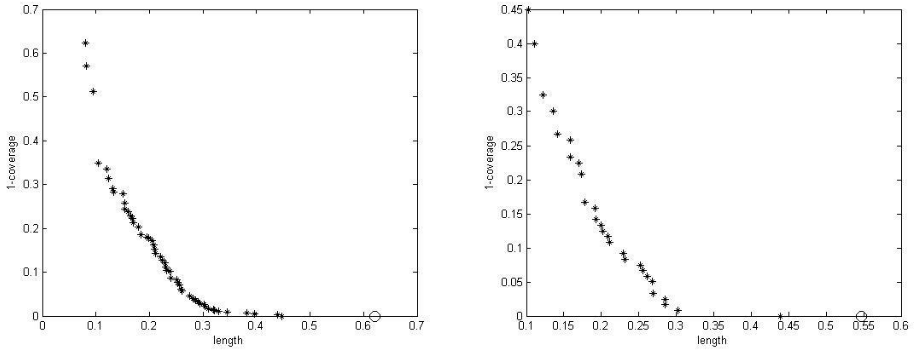
The bounds of the output produced by the granular neural network are illustrated in Figure 63. The plots are shown for the two selected values of ε.
Considering the fourth protocol, C_4, we visualize the obtained the Pareto front; see Figure 64.



(a) (b)

Figure 64. Pareto front for cases (a) and (b) and the level of granularity ε = 0.035

The circles shown in the Figures 64 and 65 are the results obtained when running a single objective PSO. In both cases they are located close or on the Pareto. The average distance between points on Pareto front and the origin of the coordinates is 0.4228 for case (a) and 0.4359 for case (b). For comparison, in Figure 65 the corresponding Pareto fronts are also shown in case when ε =0.10.



(a) (b)

Figure 65. Pareto fronts for cases (a) and (b) and ε = 0.1

When ε =0.10, the result of single objective optimization produces less specific (broader intervals), which comes with the higher coverage value. The average distance between the points on the Pareto front and the origin is 0.3142 and 0.2898 for cases (a) and (b), respectively.

It is interesting to look at the computing overhead. As an example, let us take ε =0.02.For the computing time when using C_1, C_2, C_3, and C_4 with the parameters of PSO set as before (processor: 2.50 GHz), we obtain:

case (a): T_{c1} = 0.06s, T_{c2} = 6.33s, T_{c3} = 165.40s, T_{c4} = 330.32s, case (b): T_{c1} = 0.04s, T_{c2} = 3.96s, T_{c3} = 41.96s, and T_{c4} = 84.00s.

Auto MPG data set

The random split of data results in 314 data forming a training set and the remaining 78 data used for testing. The results of the development of the neural network are summarized in Figure 66. The number of neurons in the hidden layer is set to be equal to 9.

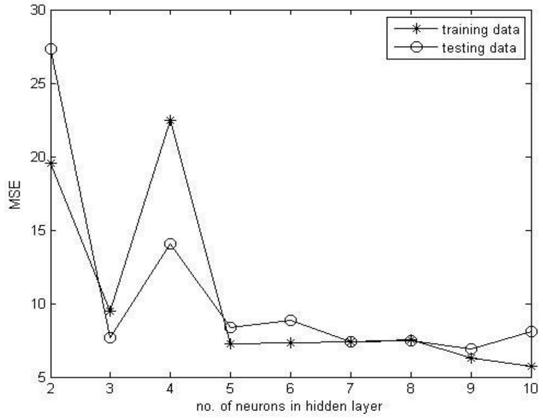


Figure 66. Performance index MSE versus the size of the hidden layer

As before, we look at cases (a) and (b).

case (a):
The values of the performance index shown in Figure 67 show a saturation effect for higher values of the level of granularity. As before, it is apparent that C_3 and C_4 produces higher coverage than the two first allocation protocols.

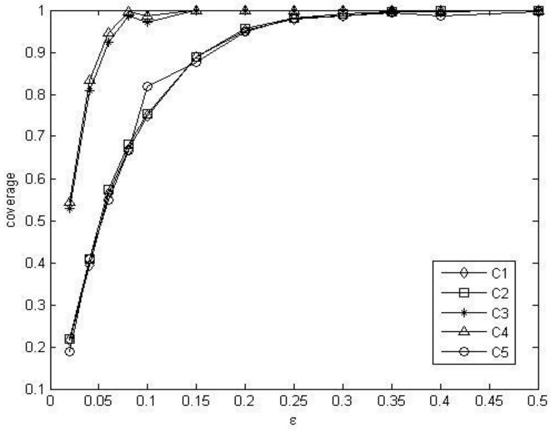


Figure 67. Coverage as a function of ε

The corresponding values of the AUC on training data are: C_1, AUC = 0.9280, C_2, AUC = 0.9297, C_3, AUC =0.9740, C_4, AUC = 0.9760, C_5, AUC = 0.9279.

For illustration, a plot of the granular neural network is included in Figure 68. It is noticeable that there is a substantial difference in the granularity of the connections where some of the connections are quite visibly associated with the broader intervals.

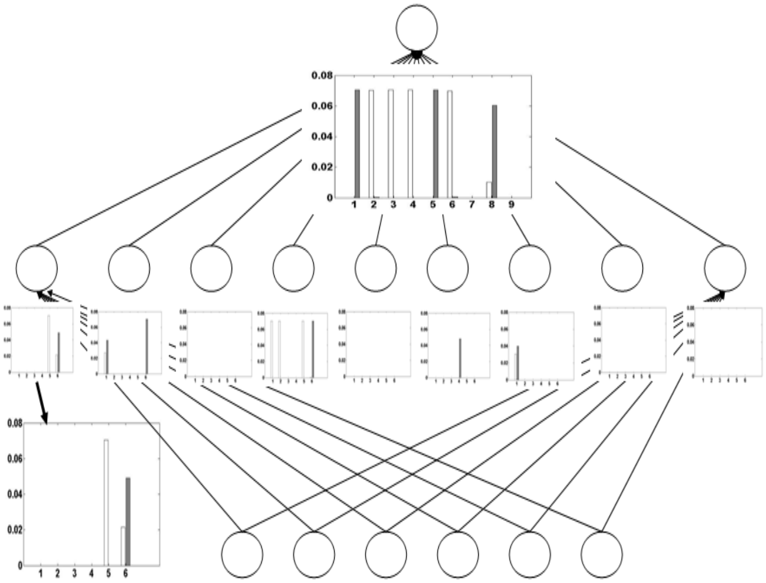


Figure 68. A topology of the granular neural network for the granularity level of 0.02 (ε) and the allocation protocol of C_4

case (b)

The corresponding values of the AUC on training data are: C_1, AUC = 0.9233, C_2, AUC = 0.9288, C_3, AUC =0.9735, C_4, AUC = 0.9756, C_5, AUC = 0.9249, refer to Figure 69.

The granular network generated for the level granularity of 0.02 is shown in Figure 70. We note that finally there are some numerical connections.

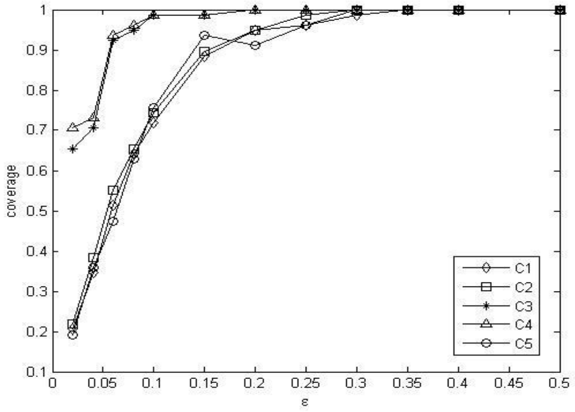


Figure 69. Coverage as a function of ε

The circles visible in Figure 71 show the PSO-developed result of the single objective. We note that this time the result of this single objective optimization locates near the edge of Pareto front for both scenarios. The average distance between points on Pareto front and original point is 3.2490 - case (a), and 3.0324 - case (b).

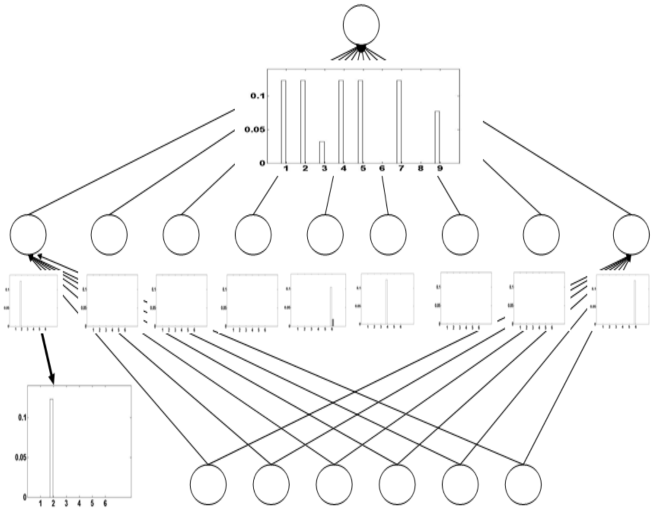
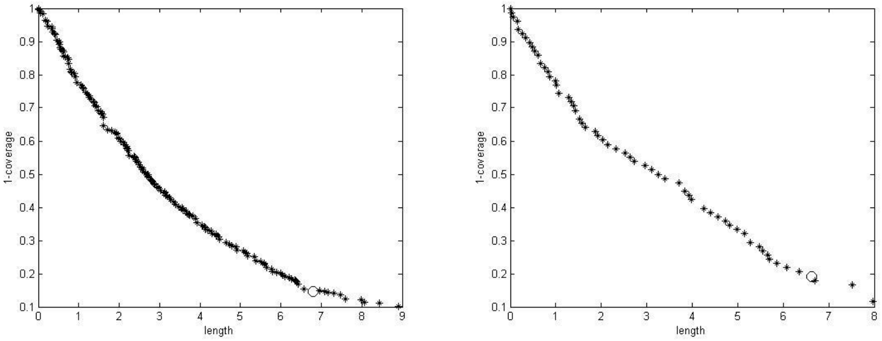


Figure 70. The optimized allocation of granularity of level 0.02 (ε) of C_4



(a) (b)

Figure 71. Pareto fronts: (a) – case (a), (b)- case (b) ε = 0.035

The actual running time of the two scenarios are: case (a): T_{c1} = 0.06s, T_{c2} = 6.32s, T_{c3} = 162.88s, T_{c4} = 325.92s, case (b): T_{c1} = 0.04s, T_{c2} = 3.95s, T_{c3} = 42.21s, T_{c4} = 84.30s.

We also experimented with the three other data sets, Boston Housing data, Bodyfat data, and PM10 data. Table 14 contains the AUC values for the optimized distribution of granularity governed by the *coverage* objective function obtained on the training data. As it could have been anticipated, the highest values of AUC are obtained when running protocol C_4 (which offers the highest level of flexibility).

Table 14. AUC values of the *coverage* objective function obtained for different protocols of allocation of information granularity for the training data and testing data (entries of the table include pairs of results - AUC training data /AUC testing data) of case (a)

protocol data	C_1	C_2	C_3	C_4	C_5
synthetic	0.9733/ 0.9732	0.9746/ 0.9727	0.9872/ 0.9865	**0.9875/ 0.9298**	0.9735/ 0.9725
auto MPG	0.8860/ 0.8890	0.8950/ 0.8940	0.9627/ 0.9617	**0.9683/ 0.7749**	0.8814/ 0.8881
Housing	0.7701/ 0.7624	0.8081/ 0.8039	0.9234/ 0.9192	**0.9408/ 0.6073**	0.7745/ 0.7688
Bodyfat	0.9807/ 0.9714	0.9841/ 0.9746	0.9918/ 0.9855	**0.9935/ 0.9640**	0.9766/ 0.9673
PM10	0.9066/ 0.9005	0.9097/ 0.8999	0.9655/ 0.9637	**0.9676/ 0.7086**	0.9069/ 0.9044

8.4. Conclusions

The concept of granular neural networks along with the underlying design practices opens a new perspective at the realizations of neural architectures at the higher level of abstraction. The granularity of the connections of the network (which is subject to intensive optimization) delivers the architectures and results of neurocomputing at the higher level of abstraction. We stressed the role of information granularity as an important design asset.

The obtained experimental results (when using synthetic and real-world data) quantify the performance of the granular neural network demonstrating its performance and capabilities for the single- and two-objective optimization scenarios and several protocols of allocation of information granularity.

It is worth stressing that the interval-valued connections (weights) of the networks were studied here as one of the simplest alternatives of granular constructs. The two promising generalizations worth pursuing could include fuzzy neural networks (in which case we admit fuzzy numbers built around numeric connections) and probabilistic neural networks (with the optimal allocation of granularity expressed in terms of the standard deviation of the probabilistic connections regarded as respective random variables).

9. Granularity of Information in Solving Group Decision-Making Problems

In group decision-making, information granularity is viewed as an essential asset, which offers a decision-maker a tangible level of flexibility using which some initial preferences conveyed by each individual can be adjusted with intent of reaching a higher level of consensus. This study is concerned with an extension of the well-known analytic hierarchy process (AHP) to the group decision-making scenario. More specifically, the admitted level of granularity gives rise to a *granular* matrix of pairwise comparisons. The granular entries represented e.g., by intervals or fuzzy sets, supply a required flexibility using that we select the most suitable numeric representative of the reciprocal matrix. The proposed concept of granular reciprocal matrices is used to optimize a performance index. The flexibility offered by the level of granularity is used to increase the level of consensus within the group. A number of numeric studies are provided to illustrate an essence of the method.

9.1. Granularity and Group Decision-Making

We have been witnessing a highly diversified plethora of approaches to multi-person or group decision-making processes realized in the setting of fuzzy sets. From the very inception of fuzzy sets, they gave rise to numerous concepts, methodologies, algorithms and applications. The studies by Zadeh et al. (Zadeh, Is there a need for fuzzy logic? 2008) offer a profound testimony to the relevance of fuzzy sets in diverse tasks of decision-making, which are inherently associated with the concept of information granularity. Multi-person or group decision-making mechanisms exploiting effective ways of forming consensus are on the agenda of fuzzy decision-making.

Some of these studies build upon the well-known established techniques being formed for a single decision maker. One of the approaches is an Analytic Hierarchy Process introduced by Saaty (Saaty, Introduction to a modeling of social decision process 1983). Since its introduction there have been a number of studies dealing with its refinements, analysis, generalizations and applications (Aguilar-Lasserre, et al. 2009) (Altuzarra, Moreno-Jimenez and Salvador 2007) (Beynon 2005) (Cakir 2008) (Jeonghwan, et al. 2010) (Kangmao, Wang and Chun 2005) (Korpela and Tuominen 1996) (McCauley-Bell and Badiru 1996) (Mustafa and Al-Bahar 1991) (Szczypinska and Piotrowski 2009) (Tung and Tang 1998). As one of the generalizations, the AHP method has been augmented to the environment of group decision-making, cf. (Dong, Zhang, et al. 2010) (Frei and Harker 1999) (Herrera-Viedma, Martinez, et al. 2005) (Herrera-Viedma, Alonso, et al. 2007) (Mata, Martinez and Herrera-Viedma 2009). A quick browsing of the existing studies in this setting helps us establish a general taxonomy of approaches with two dominant directions:

Aggregation of individual reciprocal matrices (individual judgments). Here the focus is on forming an aggregate of the reciprocal matrices. For instance, a weighted geometric mean method is used as a vehicle to realize this aggregation.
Aggregation of individual vectors of preferences produced by the AHP method. The preferences obtained on a basis of the individual reciprocal matrices are then subject to a certain aggregation.

There have been a great deal of studies elaborating on the advantages and associated limitations of the aggregation methods coming from these two categories, see (Dong, Zhang, et al. 2010) (Musilek, Guanlao and Barreiro, Genetic programming of fuzzy aggregation operations 2005).

An interesting and practically legitimate direction of promising investigations is devoted to group decision-making involving linguistic or generally speaking, non-numeric representations of assessments of individual alternatives, cf. (Dong, Hong, et al. 2011) (Herrera, Herrera-Viedma and Verdegay, A rational consensus model in group decision making using linguistic assessments 1997) (Xu and Yager 2010). In all these situations it is assumed that such types of representation are reflective of the very nature of knowledge about the decision problem. It is also assumed that such non-numeric quantifications are provided by the decision-maker.

One can convincingly note that reaching consensus requires flexibility and willingness on a part of each member of the group to adjust his/her original position. In the AHP model, these changes can be articulated through modifications of the entries of individual reciprocal matrices. Intuitively, if we allow any decision-maker to treat the entries of the reciprocal matrices not as single numeric values (which are rigid) but rather as information granules, this will bring a badly required factor of flexibility. In this manner the decision-maker may admit all numeric realizations of the reciprocal matrix, which are compatible with the more general granular abstraction such as the assumed granular matrix of pairwise comparison. We introduce a concept of granular reciprocal matrices and emphasize a role of information granularity being regarded here as an important conceptual and computational resource which can be exploited as a means to elevate the level of consensus to be reached. By admitting a certain level of information granularity, we are provided with a unique possibility to navigate in the space of priorities assigned to individual alternatives. In a very descriptive and informal way, the essence of the information granularity is to support reaching consensus through bringing some flexibility and exploiting it to the fullest possible extent. In some sense, this study is parallel to some iterative schemes of consensus building presented in (Herrera-Viedma, Martinez, et al. 2005) (Herrera-Viedma, Alonso, et al. 2007).

9.2. Building Consensus through Allocation of Information Granularity

Consider a group decision-making scenario in which there are “c” decision-makers and each of them comes with own preferences (preference vectors), **e**[1], **e**[2],…, **e**[c] obtained by running the AHP for the corresponding reciprocal matrix R[1], R[2], …, R[c]. Each reciprocal matrix has the following expression: R = [r_{ij}], and r_{ij} = 1/r_{ji} which has been discussed in chapter 2.3. An aggregate preference vector can be formed in many different ways by taking into account various methods of aggregation of R[i]s or **e**[i]s and accounting for the corresponding levels of consistency. This issue was elaborated on in the introductory section. It is worth noting that irrespectively of the aggregation technique being used; the resulting vector of preferences is somewhat a result of a post-mortem processing in which the individual decision-makers are rather passive as to changes/adjustments of their initial positions expressed in the form of the corresponding vectors of preference. The question of aggregation is interesting per se as so many distinct scenarios can be encountered. For instance, there is a group of decision-makers who share similar position about preferences and each of them is not very consistent. A few of the members of the group who are in a clearly visible minority have very different preferences however their reciprocal matrices are highly consistent. How could this become reflected in the aggregation mechanism? What if there are several sub-groups of decision-makers and each group is quite consistent in articulating their preferences but these preferences vary substantially from one sub-group to another?

Building consensus calls for some flexibility exhibited by all decision-makers who in the name of cooperative pursuits give up their initial positions and show a certain level of elasticity. They can demonstrate this by changing the original entries of the individual reciprocal matrices. The convincing generalization of the optimization problem discussed above is to optimize allocation of granularity to the individual decision-makers.

9.2.1. The objective function

It is advantageous to look at the way in which the overall process of forming consensus can be structured. Figure 72 shows a characterization of the preferences at the level of the individual decision-maker and a way in which an entire process of consensus building becomes realized at the group level.

In the interval-valued granular model of reciprocal matrices, we consider that the individual decision-maker feels equally comfortable to choose any reciprocal matrix whose values are located within the bounds established by the predetermined level of granularity. If a certain level of granularity (α) is allowed in each reciprocal matrix, it can be used in two ways to reach consensus.

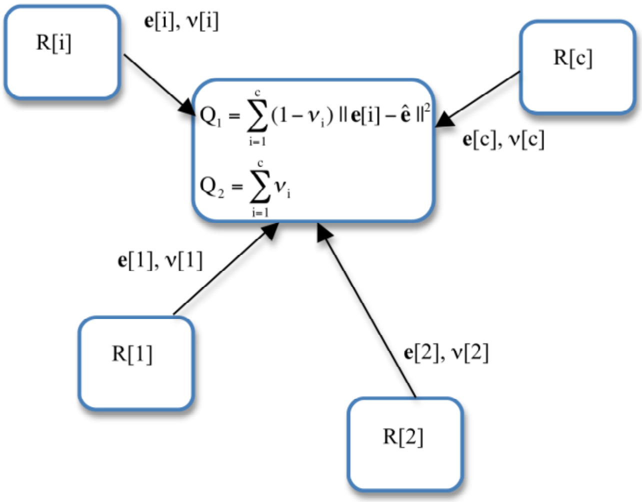

R[i]
e[i], ν[i]
R[c]
$Q_1 = \sum_{i=1}^{c}(1-\nu_i)\,\|\mathbf{e}[i]-\hat{\mathbf{e}}\|^2$
$Q_2 = \sum_{i=1}^{c}\nu_i$
e[c], ν[c]
e[1], ν[1]
R[1]
e[2], ν[2]
R[2]

Figure 72. Characterization of preferences at the level of individual decision-makers and a way of forming the objective function; see a detailed description in the text

First, we try to bring all preferences close to each other and this goal is realized by adjusting the reciprocal matrices within the bounds offered by the admissible level of granularity provided to each decision-maker.

$$Q_1 = \sum_{i=1}^{c}(1-\nu_i)\|\mathbf{e}[i]-\hat{\mathbf{e}}\|^2 \qquad (9.1)$$

where $\hat{\mathbf{e}}$ stands for the vector of preferences which minimizes the weighted sum of differences ||.|| between **e**[i] and $\hat{\mathbf{e}}$. The detailed calculations depend on the form of the distance function used here. In particular, if we select the Euclidean distance, then the optimal vector of preferences $\hat{\mathbf{e}}$ is expressed as follows

$$\hat{\mathbf{e}} = \frac{\sum_{i=1}^{c}(1-\nu_i)\mathbf{e}_i}{\sum_{i=1}^{c}(1-\nu_i)} \qquad (9.2)$$

Second, we try to increase the consistency of the reciprocal matrices and this improvement is realized at the level of individual decision-maker. The following performance index quantifies this effect

$$Q_2 = \sum_{i=1}^{c}\nu_i \qquad (9.3)$$

These are the two objectives to be minimized. If we consider the scalar version of the optimization problem, it arises in the following form

$$Q = AQ_1 + Q_2 \qquad (9.4)$$

where $A \geq 0$. The overall optimization problem reads now as follows

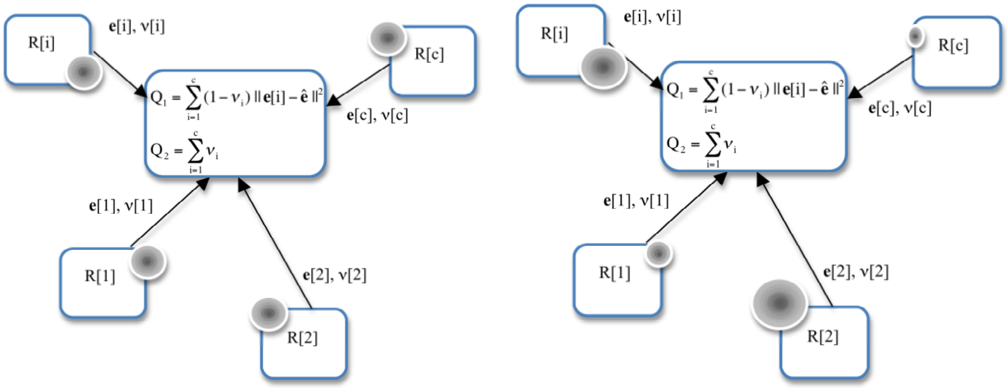
$$\text{Min}_{\,R[1], R[2], \ldots, R[c] \,\in \boldsymbol{P}(R)} Q \qquad (9.5)$$

The minimization above is carried out for all reciprocal matrices admissible because of the introduced level of information granularity α.

This optimization task is not an easy one. In virtue of the nature of the indirect relationship between optimized reciprocal matrices (which are selected from a quite large search space formed by ***P***(R)) and this calls for the use of more advanced techniques of global optimization such as e.g., evolutionary optimization, Particle Swarm Optimization (PSO), simulated annealing, ant colonies and alike.

The selection of numeric values of the weight factor A and the level of granularity α requires some clarification. The role of A is quite straightforward: by choosing its value we set up a tradeoff between the consistency obtained at the local (individual decision-maker) and global group level. The higher the value of A, the more attention is being paid to the consistency aimed at the group level. In the limit where A = 0, we are concerned with the consistency achieved at the local level only. The admissible level of granularity brings flexibility, which has to be effectively utilized – the higher the values of α, the higher the potential to reach a significant level of consistency. There is a potential of producing some quite inconsistent reciprocal matrices at the local level.

While in the above scenario we considered that the same level of granularity α is assigned across all R[i]s, these levels can be optimized as well so that each decision-maker might have an individual value of α_i that becomes available to his disposal. We require that the overall balance of granularity is retained meaning that (4.1) is satisfied.


R[i]
e[i], v[i]
R[c]
$Q_1 = \sum_{i=1}^{c}(1-v_i)\,\|\mathbf{e}[i]-\hat{\mathbf{e}}\|^2$
$Q_2 = \sum_{i=1}^{c} v_i$
e[c], v[c]
e[1], v[1]
R[1]
e[2], v[2]
R[2]
R[i]
e[i], v[i]
R[c]
$Q_1 = \sum_{i=1}^{c}(1-v_i)\,\|\mathbf{e}[i]-\hat{\mathbf{e}}\|^2$
$Q_2 = \sum_{i=1}^{c} v_i$
e[c], v[c]
e[1], v[1]
R[1]
e[2], v[2]
R[2]

(a) (b)

Figure 73. Endowing the reciprocal matrices with flexibility represented in the form of information granularity: (a) equal allocation of granularity – level of granularity set to α, and (b) optimal allocation of granularity with different level α_i assigned to individual reciprocal matrices under the constraint of overall level of granularity expressed by (4.1). Shadowed shapes located next to the reciprocal matrices underline the granular nature of these constructs

The values of α_1, α_2, …, α_h become a part of the overall optimization problem, that is (9.5) is expanded and reads now as

$$\text{Min}_{\,R[1],\, R[2],\, \ldots, R[h]\ \in \boldsymbol{P}(R)} Q$$

subject to the granularity constraint (4.1).

Figure 73 visualizes the essential role of information granularity in the endowing the reciprocal matrices with flexibility.

9.2.2. PSO Environment in Optimization of Reciprocal Matrices

The optimization of the reciprocal matrices coming from the space of granular reciprocal matrices (more precisely, interval-valued reciprocal matrices) is realized by means of the Particle Swarm Optimization (PSO), which occurs to a viable optimization alternative for this problem. The PSO is well documented in the existing literature with numerous modifications and augmentations. The reader may refer to the generic flow of computing in which velocities and positions of the particles are updated we discussed in chapter 2.4. What is important in this setting is a formation of the particle whose entries are transformed to the values of the reciprocal matrix. From the perspective of the PSO itself, all entries of the particle are located in the unit interval. As the matrix exhibits the reciprocity property, only one of the entries located symmetrically with respect to the main diagonal has to be optimized by the PSO and the other one is determined on a basis of the reciprocity property.

Let us assume that the entry of interest of the particle is “x”. It is transformed linearly according to the expression, $z = a + (b-a)x$. a and b are lower and upper bounds of an interval which has been discussed about its formalism in chapter 4.4. We compute its inverse, $1/z$, and round it off to the nearest integer in the set $\{1, \ldots, 9\}$. Say, the result is the integer value z_0. To retain reciprocality property, we calculate the inverse of z_0 and replace the result of the linear transformation (z) from which we have started with. The essence of this scheme is illustrated in Figure 74.

For example, consider that r_{ij} is equal to 1/5, the admissible level of granularity α =0.05 and the corresponding entry of the particle is x =0.3. Then the corresponding interval of the granular pairwise comparison matrix computed as given by (4.7) becomes equal to [a, b] = [0.155, 0.244]. Subsequently z = 0.181. This gives rise to the integer value of the (j, i)-th entry of R positioned symmetrically equal to 6 and finally the modified value of r_{ij} becomes equal to 1/6.

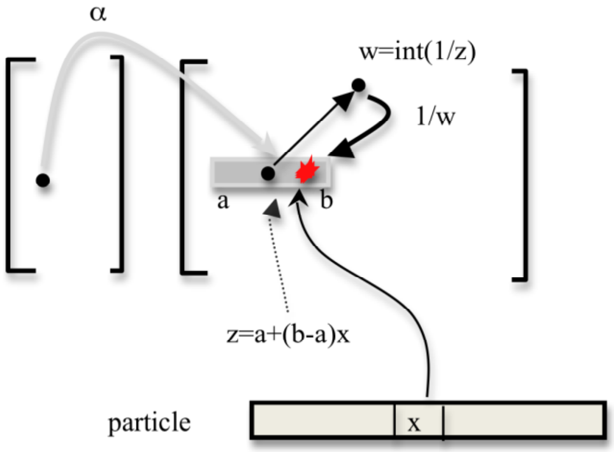


Figure 74. Formation of the reciprocal matrix with the use of the PSO and predetermined level of granularity α; a and b are two bounds of an interval

The overall particle is composed of the individual segments where each of them is concerned with the optimization of the parameters of the reciprocal matrices. If the allocation of granularity is also optimized then the values of α_is are also optimized. These optimized values are then used in the decoding of the content of the corresponding segments of the particle by calibrating the intervals of admissible granularity (see 4.8).

9.3. Experimental Studies

In this section, we illustrate the method and highlight its main features by presenting several examples. Here we use a version of the PSO as discussed in chapter 2.4 (B) (Kennedy and Eberhart, Particle swarm optimization 1995). The size of the swarm consists of 1,000 particles and the algorithm was run for 200 generations. Since the search space is quite large (there are around 50 elements in the particle), it has been found that this particular size of the population was suitable to realize a search process.

Example 1 We consider five reciprocal matrices coming from five decision-makers. The entries of these matrices are reflective of the pairwise comparisons of five alternatives (options).

$$R_1 = \begin{bmatrix} 1 & 1/4 & 1/3 & 1/7 & 1/3 \\ 4 & 1 & 1/6 & 1/5 & 1/6 \\ 3 & 6 & 1 & 1/2 & 1/3 \\ 7 & 5 & 2 & 1 & 1/2 \\ 3 & 6 & 3 & 2 & 1 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 1/2 & 1/4 & 1/3 & 1/2 \\ 2 & 1 & 1/5 & 1/2 & 2 \\ 4 & 5 & 1 & 1 & 1/2 \\ 3 & 2 & 1 & 1 & 3 \\ 2 & 1/2 & 2 & 1/3 & 1 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 1 & 1/2 & 1/2 & 1/8 & 1/8 \\ 2 & 1 & 1/4 & 1/3 & 1/9 \\ 2 & 4 & 1 & 1/5 & 1/5 \\ 8 & 3 & 5 & 1 & 1/4 \\ 8 & 9 & 5 & 4 & 1 \end{bmatrix} \qquad R_4 = \begin{bmatrix} 1 & 1/2 & 1/3 & 1/7 & 1/2 \\ 2 & 1 & 1/5 & 1/2 & 1/3 \\ 3 & 5 & 1 & 1/4 & 1/4 \\ 7 & 2 & 4 & 1 & 1 \\ 2 & 3 & 4 & 1 & 1 \end{bmatrix}$$

$$R_5 = \begin{bmatrix} 1 & 1 & 3 & 5 & 5 \\ 1 & 1 & 1/3 & 4 & 8 \\ 1/3 & 3 & 1 & 3 & 6 \\ 1/5 & 1/4 & 1/3 & 1 & 6 \\ 1/5 & 1/8 & 1/6 & 1/6 & 1 \end{bmatrix}$$

The corresponding maximal eigenvalues of the reciprocal matrices are equal to: λ_1=5.68, λ_2=5.79, λ_3=5.46, λ_4=5.73, λ_5=5.70, which give rise to the following values of the inconsistency index: ν_1=0.171, ν_2=0.198, ν_3=0.114, ν_4=0.183, ν_5=0.175.

It is noticeable that all the matrices exhibit a similar level of consistency with an exception of the reciprocal matrix R_3, whose consistency level is higher than the rest of the matrices. The corresponding eigenvectors are given below

$$\mathbf{e}_1 = [0.141 \quad 0.206 \quad 0.506 \quad 0.753 \quad 1.000]^T$$
$$\mathbf{e}_2 = [0.239 \quad 0.509 \quad 1.000 \quad 0.970 \quad 0.638]^T$$
$$\mathbf{e}_3 = [0.075 \quad 0.106 \quad 0.195 \quad 0.481 \quad 1.000]^T$$
$$\mathbf{e}_4 = [0.187 \quad 0.268 \quad 0.501 \quad 1.000 \quad 0.884]^T$$
$$e_5 = [1.000 \quad 0.647 \quad 0.768 \quad 0.254 \quad 0.096]^T$$

There is a diversity of preferences discovered here: the fifth alternative is picked up twice and the remaining alternatives are identified once. In case no granularity is admitted, the aggregated fuzzy set of preferences computed by making use of (9.2) comes with a vector of membership values

$$\hat{\mathbf{e}} = [0.3255 \;\; 0.3427 \;\; 0.5861 \;\; 0.6867 \;\; 0.728]^T$$

and here the fifth alternative is characterized by the highest membership degree. The plots of the decision profiles produced by the individual decision-makers are illustrated in Figure 75.

Before proceeding with the PSO optimization of the reciprocal matrices when supplied with the required granularity level, it becomes instructive to analyze an impact of the allocated level of granularity on the possible changes in terms of the improvement or deterioration of consistency of the matrices. To quantify the effect of imposed granularity, for a given reciprocal matrix R we allow a certain value of α, for this specific value randomly generate a reciprocal matrix coming from a granular representation of R, ***P***(R), and compute the associated inconsistency index. The calculations are repeated 500 times for each value of α. The corresponding plots of ν versus α are shown in Figure 76. In addition, in these figures we visualize average values of the inconsistency index.

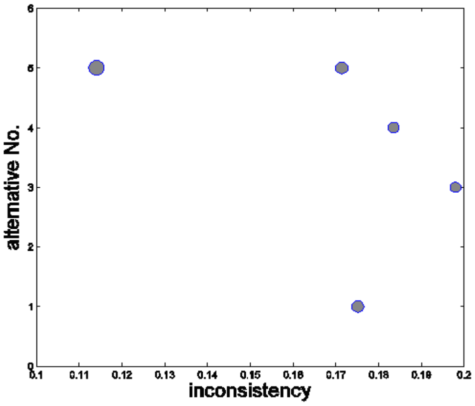
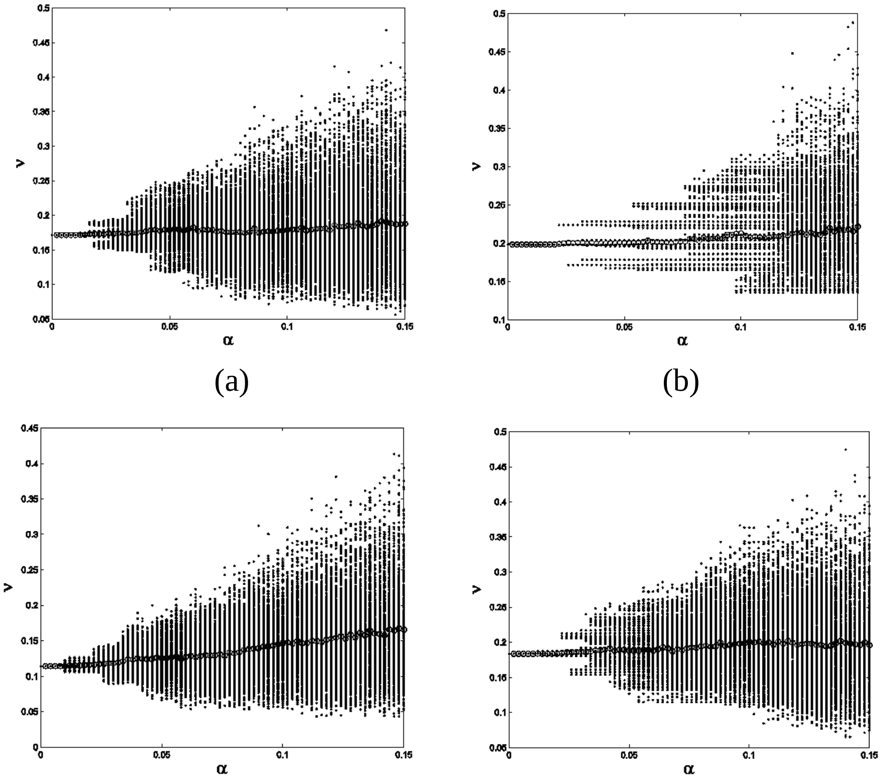


Figure 75. Decision profiles generated by the individual reciprocal matrices

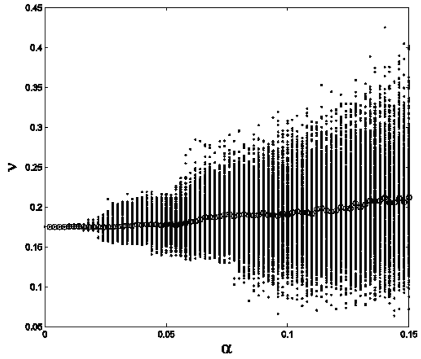


(a)

(b)

(c)

(d)



(e)

Figure 76. Plots of inconsistency index ν versus α for reciprocal matrices R_1-R_5: (a)-(e)

When increasing the values of α, the possibility of arriving at more consistent reciprocal matrix increases (which is not surprising as we have introduced some level of flexibility that we intend to take advantage of) however the likelihood of producing a very inconsistent matrix increases as well. The average value of inconsistency remains pretty steady with respect to the increasing values of ε as reported for all matrices however there is some slight upward trend for higher value of α.

Considering a level of granularity (α) (which is the same for all reciprocal matrices), we run the optimization of the entries of the reciprocal matrices. The performance of the PSO is illustrated in Figure 77; the most significant improvement is noted at the very beginning of the optimization and afterwards there is a clearly visible stabilization where the values of the fitness function remain steady.

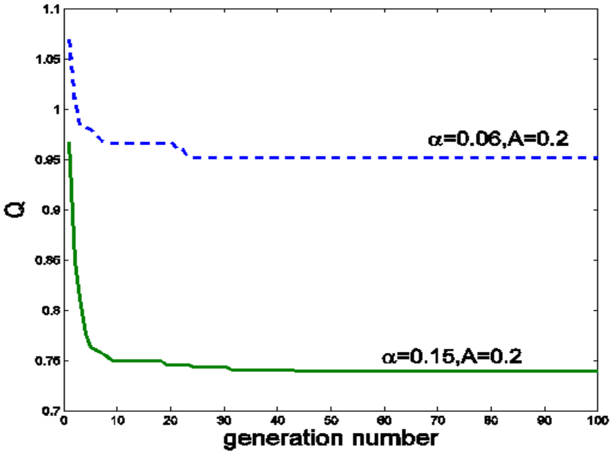
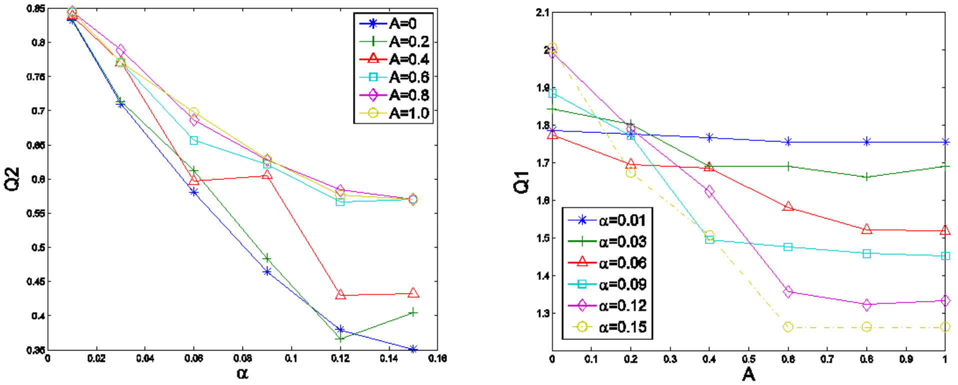


Figure 77. Values of fitness function Q in consecutive generations of the PSO for selected values of α = 0.06 and 0.15 (here A =0.2).

Let us investigate an impact of the granularity level (α) and the weight factor A in the composite fitness function on the performance of the method and the form of the obtained results. For A = 0, the optimization involves each of the reciprocal matrices individually. Here the increase in the values of ε offers more flexibility, which if prudently used (optimized by the PSO), produces the reciprocal matrices of higher consistency. This effect is clearly visible in Figure 78 (the curve for A=0). The advantageous effect of granularity is visible: with the increasing values of α, the reciprocal matrices become more flexible and this translates into higher levels of consistency of the evaluations of the alternatives. The same effect is visible for nonzero values of A: if there is some interaction, the positive impact of introduced granularity is beneficial (the overall level of inconsistency quantified by Q_2 is a decreasing function of α). The strictly monotonic character of this relationship is not retained for higher values of A as again illustrated in Figure 78 for A=1.0. This is not surprising, however, as the performance index optimized by PSO is not Q_2 itself but Q and as such, this composite index includes also the effect of consistency of the preferences themselves.



(a) (b)

Figure 78. Plot of Q_2 versus α and Plots of Q_1 versus A shown for selected values of α

Figure 79 includes a number of plots of Q_1 (reflective of the consistency among preferences) regarded as functions of A for selected levels of granularity. Again, the advantageous effect of granularity is visible as higher values of ε result in lower values of Q_1. For the fixed value of α, there is a clearly visible saturation effect: when moving beyond a certain point, further increase of values of A does not reduce the values of Q_1.

The decision-making profiles obtained for the granular reciprocal matrices for selected combinations of the values of A and ε are included in Figure 79. It becomes apparent that different combinations of these two parameters result in several decision-making scenarios pointing at specific alternatives as well as the decision profiles.

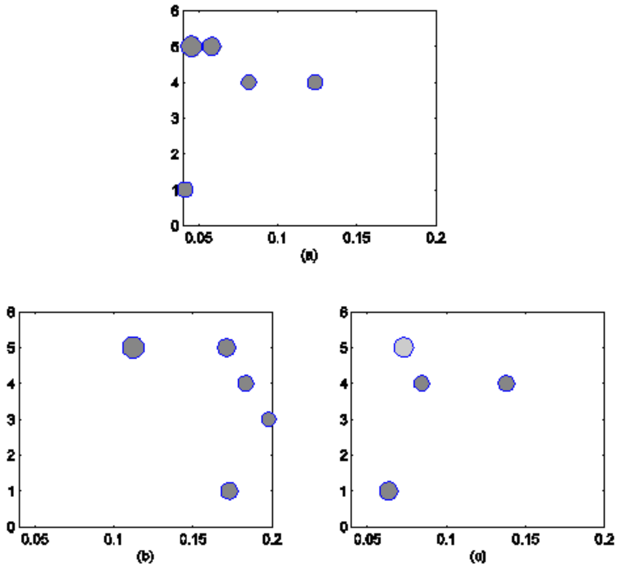
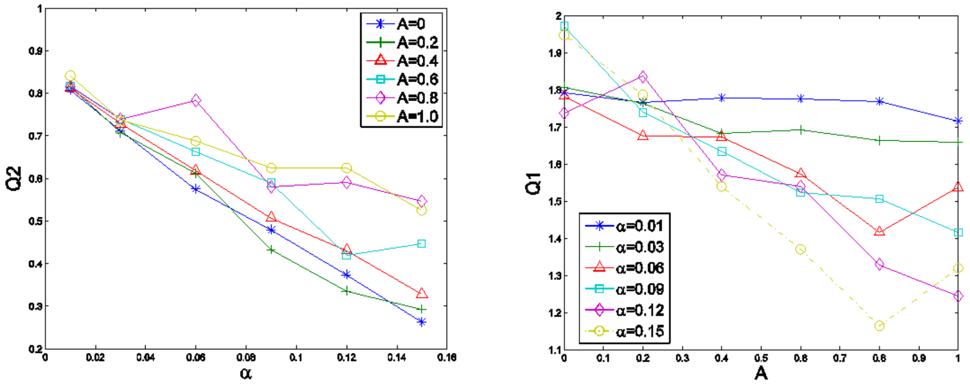


Figure 79. Decision profiles generated by the individual reciprocal matrices: (a) A= 0, α=0.15 (b) A= 0.4, α=0.01 (c) A= 0.4, α=0.15

For α = 0.15, as shown in Figure 79 (a) the level of consistency has been significantly reduced which speaks to the important role played by information granularity. For the last example, Figure 79 (c) both nonzero values of the parameters A and α both the level of consistency as well as the agreement in the achieved preferences have been increased. The decision profiles are shown in Figure 81.



(a) (b)

Figure 80. Q_2 versus α; given are selected values of A and Plots of Q_1 versus A for selected values of α

The plots of fuzzy sets of preferences for the selected values of the parameters are included in Figure 82. It is noticeable that nonzero values of A and ε make fuzzy sets of preference to become positioned close to each other, which stands in contrast with the situation displayed in Figure 82 (a).

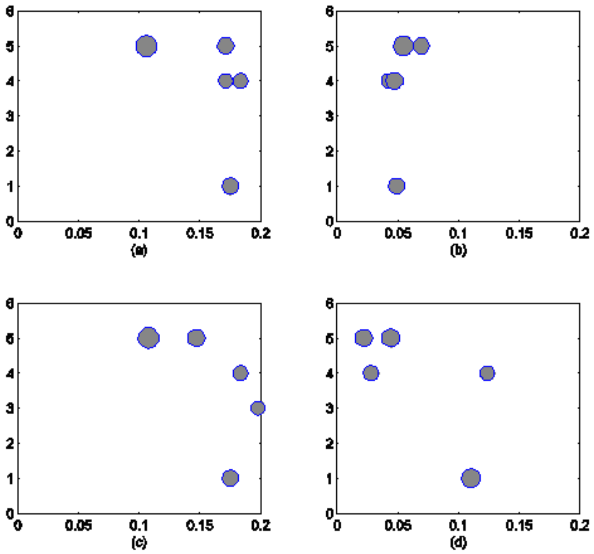
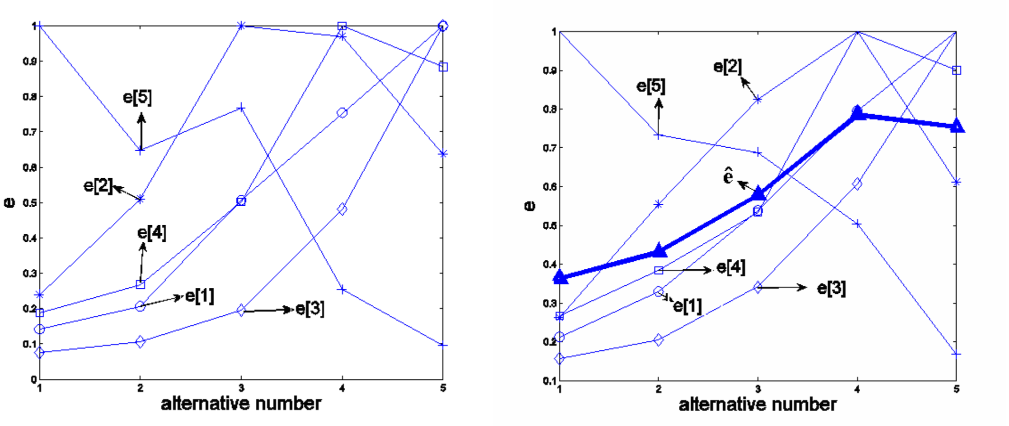


Figure 81. Decision profiles generated by the individual reciprocal matrices: (a) A= 0, α=0.01 (b) A= 0, α=0.15 (c) A= 0.4, α=0.01 (d) A= 0.4, α=0.15



(a) (b)

Figure 82. Fuzzy sets of preferences e[1], e[2],…, e[5] and the result of aggregation $\hat{e}$: (a) A= 0, α=0 (b) A= 0.8, α=0.15

So far, we have allocated the same level of granularity to all reciprocal matrices. Now the available level of granularity is allocated to all matrices when the allocation itself is optimized by the PSO along with the optimization of the entries of the individual R$_i$s. In this case, the obtained plots of Q_2 and Q_1 are shown in Figure 80 while some selected decision profiles are included in Figure 81.

As the PSO was used to optimize Q, it is helpful to evaluate how much the optimized allocation of granularity becomes reflected in the reduction of the values of Q in contrast to the values of Q where each reciprocal matrix is affected at the same level. The summary of the results is presented in Table 15. We notice that in all cases some improvement is present with the achieved level of improvement as high as 25%.

Table 15. Values of optimized performance index Q for the same value of α and optimized allocation of granularity for selected values of α and A (each entry of the table includes two numbers Q/Q'–without granularity allocation (Q) and with granularity allocation (Q')

A α	0	0.2	0.4	0.6	0.8	1.0
0.01	0.83/0.81	1.19/1.17	1.54/1.52	1.9/1.88	2.25/2.23	2.6/2.56
0.03	0.71/0.71	1.07/1.06	1.45/1.4	1.78/1.75	2.12/2.07	2.46/2.39
0.06	0.58/0.57	0.95/0.95	1.27/1.29	1.61/1.61	1.9/1.92	2.21/2.23
0.09	0.46/0.48	0.84/0.78	1.2/1.16	1.51/1.5	1.79/1.79	2.08/2.04
0.12	0.38/0.37	0.72/0.7	1.08/1.06	1.38/1.34	1.64/1.65	1.91/1.87
0.15	0.35/0.26	0.74/0.65	1.03/0.94	1.33/1.27	1.58/1.48	1.83/1.84

Example 2. Here we consider a larger group of decision-makers who have quantified the pairwise preferences of five alternatives in the following matrices

$$R_1 = \begin{bmatrix} 1 & 1/7 & 1/5 & 1/4 & 1 \\ 7 & 1 & 4 & 2 & 9 \\ 5 & 1/4 & 1 & 1/2 & 3 \\ 4 & 1/2 & 2 & 1 & 7 \\ 1 & 1/9 & 1/3 & 1/7 & 1 \end{bmatrix} \qquad R_2 = \begin{bmatrix} 1 & 9 & 7 & 6 & 1 \\ 1/9 & 1 & 1/3 & 1/2 & 1/7 \\ 1/7 & 3 & 1 & 2 & 1/5 \\ 1/6 & 2 & 1/2 & 1 & 1/6 \\ 1 & 7 & 5 & 6 & 1 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 1 & 1/6 & 1/5 & 1 & 1/7 \\ 6 & 1 & 2 & 7 & 1/2 \\ 5 & 1/2 & 1 & 9 & 1/3 \\ 1 & 1/7 & 1/9 & 1 & 1/9 \\ 7 & 2 & 3 & 9 & 1 \end{bmatrix} \qquad R_4 = \begin{bmatrix} 1 & 5 & 8 & 1/3 & 1/2 \\ 1/5 & 1 & 1 & 1/6 & 1/7 \\ 1/8 & 1 & 1 & 1/9 & 1/8 \\ 3 & 6 & 9 & 1 & 2 \\ 2 & 7 & 8 & 1/2 & 1 \end{bmatrix}$$

$$R_5 = \begin{bmatrix} 1 & 1/2 & 1/5 & 8 & 1 \\ 2 & 1 & 1/4 & 5 & 1/3 \\ 5 & 4 & 1 & 6 & 4 \\ 1/8 & 1/5 & 1/6 & 1 & 1/8 \\ 1 & 3 & 1/4 & 8 & 1 \end{bmatrix} \qquad R_6 = \begin{bmatrix} 1 & 8 & 1/7 & 1/6 & 1 \\ 1/8 & 1 & 1/8 & 1/5 & 1/2 \\ 7 & 8 & 1 & 3 & 9 \\ 6 & 5 & 1/3 & 1 & 5 \\ 1 & 2 & 1/9 & 1/5 & 1 \end{bmatrix}$$

$$R_7 = \begin{bmatrix} 1 & 1/7 & 1/6 & 1 & 2 \\ 7 & 1 & 1/3 & 5 & 3 \\ 6 & 3 & 1 & 8 & 9 \\ 1 & 1/5 & 1/8 & 1 & 6 \\ 1/2 & 1/3 & 1/9 & 1/6 & 1 \end{bmatrix} \qquad R_8 = \begin{bmatrix} 1 & 6 & 1/7 & 1/4 & 1 \\ 1/6 & 1 & 1/9 & 1/5 & 2 \\ 7 & 9 & 1 & 3 & 7 \\ 4 & 5 & 1/3 & 1 & 5 \\ 1 & 1/2 & 1/7 & 1/5 & 1 \end{bmatrix}$$

For these matrices we obtain the following set of eigenvalues and the corresponding values of the inconsistency index:

λ_1=5.1180, λ_2=5.1177, λ_3=5.1605, λ_4=5.1599, λ_5=5.5979, λ_6=5.5992, λ_7=5.5988, λ_8=5.6007; ν_1=0.0295, ν_2=0.0294, ν_3=0.0401, ν_4=0.04, ν_5=0.1495, ν_6=0.1498, ν_7=0.1497, ν_8=0.1502

along with the eigenvalues describing levels of preference assigned to alternatives

$$\mathbf{e}_1 = [0.109 \quad 1.000 \quad 0.334 \quad 0.559 \quad 0.100]^T$$

$$\mathbf{e}_2 = [1.000 \quad 0.093 \quad 0.217 \quad 0.148 \quad 0.879]^T$$

$$\mathbf{e}_3 = [0.103 \quad 0.636 \quad 0.466 \quad 0.086 \quad 1.000]^T$$

$$\mathbf{e}_4 = [0.457 \quad 0.106 \quad 0.087 \quad 1.000 \quad 0.686]^{\mathbf{T}}$$

$$e_5 = [0.282 \quad 0.280 \quad 1.000 \quad 0.067 \quad 0.416]^T$$

$$\mathbf{e}_6 = [0.196 \quad 0.069 \quad 1.000 \quad 0.533 \quad 0.120]^T$$

$$\mathbf{e}_7 = [0.129 \quad 0.527 \quad 1.000 \quad 0.187 \quad 0.083]^T$$

$$\mathbf{e}_8 = [0.212 \quad 0.100 \quad 1.000 \quad 0.484 \quad 0.109]^T$$

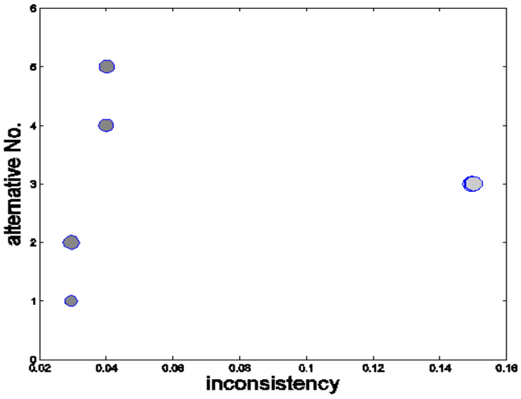
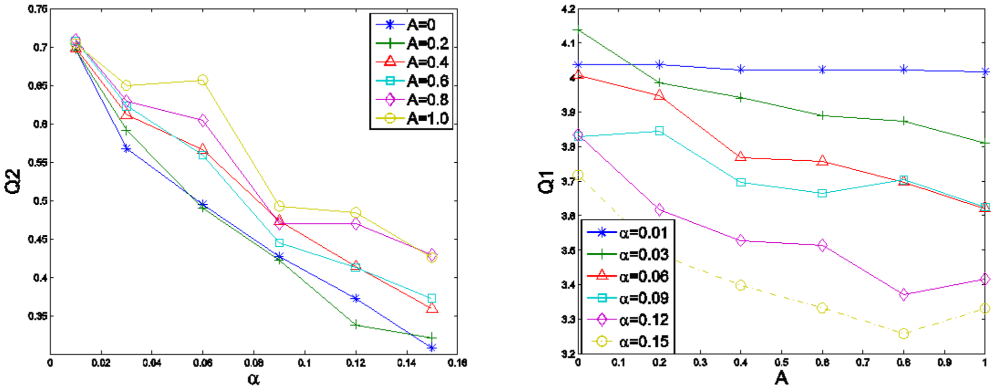


Figure 83. Decision profiles generated by the individual reciprocal matrices

The preferences are quite scattered and the two groups of decision-makers characterized by different levels of consistency are visible.

After running the PSO algorithm, the obtained results are reported in Figure 84 (a) and (b). The cumulative level of inconsistency (expressed by Q_2) drops very quickly with the increasing values of granularity, Figure 84 (a) and this effect is noticeable for different values of A. The values of the performance index Q_1 are reduced however the trend is not so visible as in case of Q_1. The decision profiles and the fuzzy sets of preference of alternatives are displayed in Figure 83 and 84.



(a) (b)

Figure 84. Q_2 versus α and Q_1 versus A

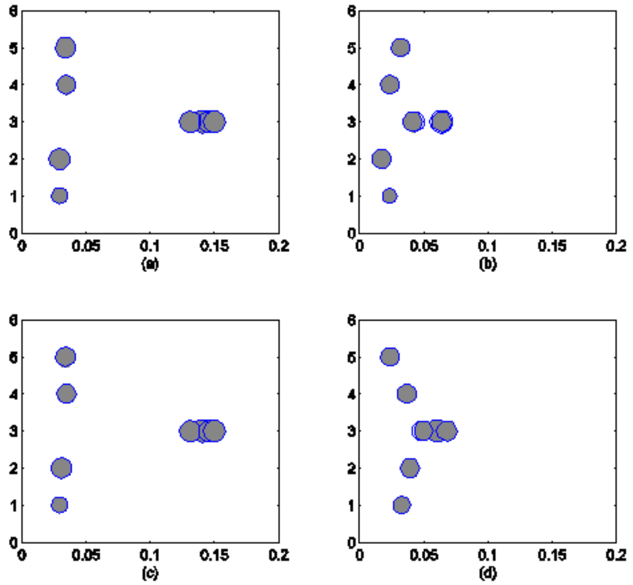
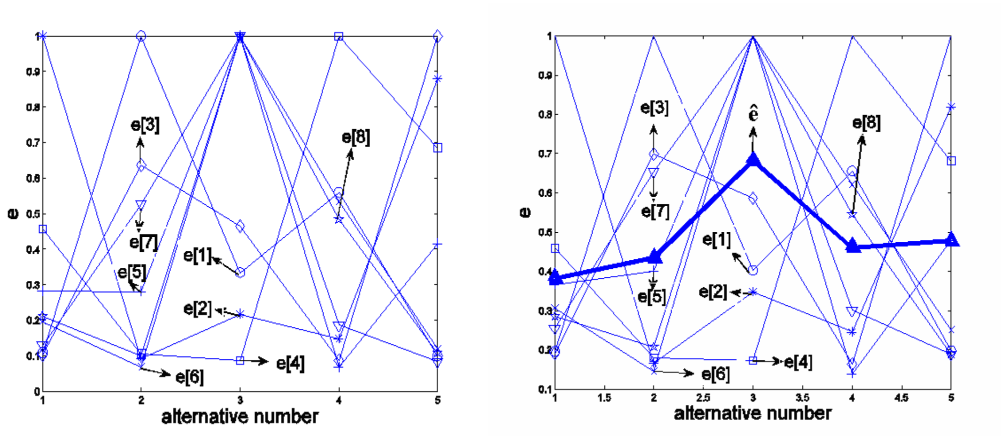


Figure 85. Decision profiles generated by the individual reciprocal matrices: (a) A= 0, α=0.01 (b) A= 0, α=0.15 (c) A= 0.4, α=0.01 (d) A= 0.4, α=0.15

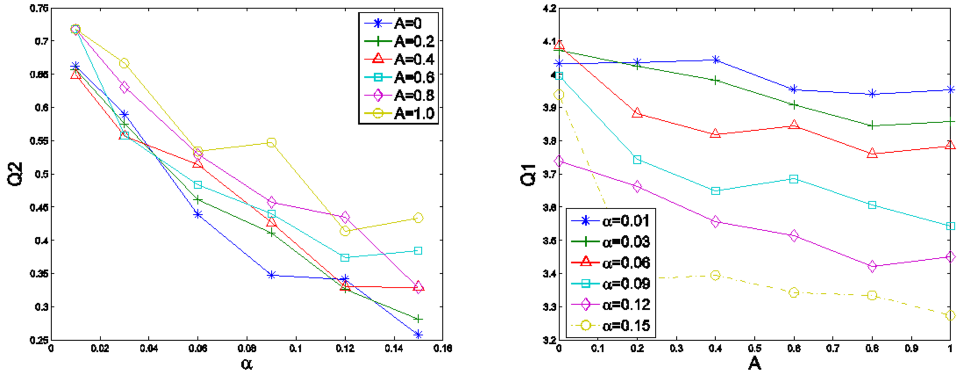
Figure 85 includes as series of decision profiles and as such offers a general view on how the preferences are affected by various values of α and A. Similarly, we show the membership functions of the fuzzy sets of preference in Figure 86.



(a) (b)

Figure 86. Fuzzy sets of preferences: (a) A= 0, α=0 (b) A= 0.8, α=0.15

After optimization of the allocation of granularity to the reciprocal matrices, the obtained results in terms of Q_1 and Q_2 are included in Figure 87 (a) and (b).



(a) (b)

Figure 87. Q_2 versus α and Q_1 versus A

After running PSO (with 1,000 particles, 100 generations), the optimized value of Q without optimization of the level of granularity (the same value of α for each R[i]) and with the optimization of α under some certain values of A and α are shown below. In each cell, the "a/b" form represent the two values of Q. "a" is the value of Q obtained when no optimization of α was done and "b" is the values of Q obtained with the optimization of α. We notice that in most cases some improvement is present however its level varies in-between -3.81% and 19 %. See Table 16.

Table 16. The values of Q for uniform and optimized allocation of the levels of granularity

A α	0	0.2	0.4	0.6	0.8	1.0
0.01	0.7/0.66	1.5/1.46	2.31/2.27	3.12/3.09	3.93/3.87	4.72/4.67
0.03	0.57/0.59	1.39/1.38	2.19/2.15	2.96/2.9	3.73/3.71	4.46/4.52
0.06	0.5/0.44	1.28/1.24	2.07/2.04	2.81/2.79	3.56/3.54	4.28/4.32
0.09	0.43/0.35	1.19/1.16	1.95/1.89	2.64/2.65	3.43/3.34	4.12/4.09
0.12	0.37/0.34	1.06/1.06	1.83/1.75	2.52/2.48	3.17/3.17	3.9/3.86
0.15	0.31/0.26	1.02/0.96	1.72/1.69	2.37/2.39	3.04/3	3.76/3.71

9.4. Conclusions

Information granularity is an important and useful asset supporting reaching consensus in group decision-making. It offers a badly needed flexibility so that the granular reciprocal matrices can produce numeric realizations so that they are both of higher consistency at the individual level and the consistency at the group level. The PSO environment has been shown to serve as an efficient optimization vehicle allocating levels of granularity to the corresponding reciprocal matrices formed by the decision-makers. The granular representation of reciprocal matrices discussed in this chapter was the one using intervals; however any other formalisms of Granular Computing, especially fuzzy sets could be equally applicable here. There is an interesting further research pursuit, which relates to the proposals provided in (Alonso, et al. 2010) (Perez, Cabrerizo and Herrera-Viedma 2010) where interesting concepts of consensus formation can be combined with the principle of justifiable granularity and its allocation presented in this study.

10. Granulation of Linguistic Information in Decision Making

To be fully utilized, linguistic information present in decision-making has to be made operational through information granulation. This chapter is concerned with information granulation present in the problems of AHP, which is available in the characterization of a pairwise assessment of alternatives studied in the decision-making problem. The granulation of entries of reciprocal matrices forming the cornerstone of the AHP is formulated as an optimization problem in which an inconsistency index is minimized by a suitable mapping of the linguistic terms on the predetermined scale. Both individual and group decision-making models of AHP are discussed.

10.1. Linguistic Information in Decision Making

In decision-making problems, we are commonly faced with information provided by humans, which is inherently non-numeric. Partial evaluations, preferences, weights are expressed linguistically. The evident role of fuzzy sets in decision-making and associated important processes such as consensus building is well documented in the literature, see (Alonso, et al. 2010) (Herrera, Herrera-Viedma and Verdegay, A rational consensus model in group decision making using linguistic assessments 1997) (Herrera-Viedma, Alonso, et al. 2007) (Perez, Cabrerizo and Herrera-Viedma 2010).

While fuzzy sets have raised awareness about the non-numeric nature of information, its importance, a need for its handling and provided a great deal of techniques of processing fuzzy sets, the fundamental issue about a transformation of available pieces of linguistic information into formal constructs of information granules. The resulting information granules are afterwards effectively processed within the computing setting pertinent to the assumed framework of information granulation. The linguistic terms such as high, medium, etc. are in common use. It is not clear, however, how they have to be translated into the entities, which can be further seamlessly processed using the formalisms of sets, fuzzy sets, rough sets and alike. Likewise, it is not straightforward what optimization criterion can be envisioned when arriving at the formalization of the linguistic terms through information granules.

Given the diversity of decision-making problems and being alerted to the fact that each of their categories could come with some underlying specificity and particular requirements, in this study we concentrate on the Analytic Hierarchy Process (AHP) model, which addresses a large and important category of decision-making schemes, see (Saaty, How to handle dependence with the analytic hierarchy process 1987) (Saaty, A new macroeconomic forecasting and policy evaluation method using the analytic hierarchy process 1987) (Wang and Chen 2008). There is a visible abundance literature on the refinements and generalizations of these models as well as various applied studies (Dong, Zhang, et al. 2010) (Dong, Hong, et al. 2011) (Jeonghwan, et al. 2010) (Korpela and Tuominen 1996) (Saaty and Hu, Ranking by eigenvector versus other methods in the analytic hierarchy process 1998).

The pairwise comparisons of alternatives are articulated in terms of linguistic quantifications, say *highly* preferred, *moderately* preferred, etc. Each term is associated with some numeric values. It has been identified quite early in the development of the AHP-like models that the single numeric values taken from the 1-9 scale do not necessarily fully reflect the complexity of the non-numeric nature of the pairwise comparisons. The first approach along this line was presented in (van Laarhoven and Pedrycz 1983) where the authors admitted triangular fuzzy numbers defined in the scale. There have been a significant number of pursuits along this line. The granular nature of the pairwise assessments was discussed in the context of a group decision-making where reaching consensus calls for some flexibility of evaluations individual assessments have to be endowed with to facilitate processes of consensus building.

In the study, we focus on two scenarios: decision-making involving a single decision –maker and decision-making realized in presence of several decision-makers (group decision-making problem). The granulation formalism being discussed in the study concerns intervals and fuzzy sets however it applies equally well to any other formal scheme of information granulation, say probabilistic sets, shadowed sets, rough sets. It is worth stressing here that information granulation offers an *operational* model of the AHP to be used in presence of linguistic pairwise comparisons. The PSO framework supporting the formation of information granules helps translate linguistic quantification into meaningful information granules so that the highest consistency of the evaluations is achieved.

The linguistic terms used in a pairwise comparison of alternatives are expressed linguistically by admitting qualitative terms. They can be organized in a linear fashion, as there is some apparent linear order among them. The terms themselves are not operational meaning that no further processing can be realized, which involves a quantification of the linguistic terms.

10.2. Quantification and Optimization of Interval Linguistic Information

The construction of information granules is realized as a certain optimization problem. In this section, we elaborate on the fitness function, its realization, and the PSO optimization along with the corresponding formation of the components of the swarm.

The objective of the fitness function is to provide a quantification of the information granules on which information granules are to be mapped. Considering the nature of the AHP model, the quality of the solution (preference vector) is expressed in terms of the inconsistency index. For the given vector of cutoff points, their quality associates with the corresponding value of the inconsistency index. The minimization of the values of the index by adjusting the positions of the cutoff points in the 1…9 scale is realized by the PSO. When it comes to the formation of the fitness function, its determination has to take into account a fact that interval-valued entries of the reciprocal matrix have to return numeric values of the fitness function. This is realized as follows. As we encounter information granules in the form of intervals, we consider a series of their realizations. Let us consider that an individual generated by the PSO has produced a collection of cutoff points specified by the individual in the swarm on which PSO operate, which is located on the [1, 9] scale. For instance, if there are three linguistic terms in the reciprocal matrix, say L (*Low*), M (*Medium*), and H (*High*), and the corresponding cutoff points are a_1 and a_2, respectively, we arrive at the mapping (representation) of the terms as follows L: [1, a_1]   M: [a_1, a_2]   H: [a_2, 9]. If we consider "m" linguistic values, this results in "m-1" cutoff points. Being arranged in a single vector, they constitute an individual in the swarm of the PSO.

A finite series of realizations of the information granules being the entries of the granular reciprocal matrix is formed by randomly generating entries coming from the above intervals, plugging them into the reciprocal matrix, and computing the largest eigenvalue and the corresponding value of the inconsistency index. The average of the values of the inconsistency index is the fitness function associated with the particle formed by the cutoff points a_1 and a_2. A way of the formation of the fitness function is in line with the standard practices encountered in Monte Carlo simulations, see (Pelaez and Lamata 2003) (Williams 1991).

Example 1. We consider a 5 by 5 reciprocal matrix with the three linguistic entries

$$R = \begin{bmatrix} 1 & M & 1/L & 1/M & 1/H \\ 1/M & 1 & 1/L & 1/H & 1/H \\ L & L & 1 & 1/L & 1/H \\ M & H & L & 1 & 1/L \\ H & H & H & L & 1 \end{bmatrix}$$

The granular matrix R is sampled 500 time (the numbers drawn from the uniform distribution defined over the corresponding subintervals of the [1, 9] scale). Recall that the fitness function is the average of the inconsistency index computed over each collection of 500 reciprocal matrices. The process of learning realized by the

PSO is illustrated in Figure 88 where we show the values of the fitness function obtained in successive generations. The parameters of the PSO were set up as follows: the number of particles is 100, and the number of iterations is set to 500. The progression of the optimization is quantified in terms of the fitness function obtained in successive generations, see Figure 88.

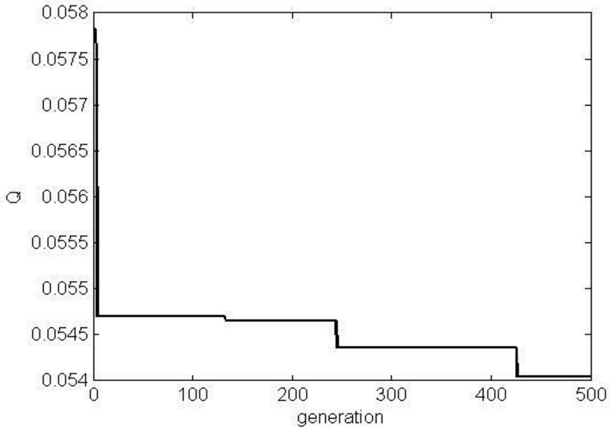


Figure 88. The values of the fitness versus generations of the PSO

To put the obtained optimization results in a certain context, we report the performance obtained when considering a uniform distribution of the cutoff points over the scale, which are equal to 3.67 and 6.34, respectively, see Figure 89. The average inconsistency index assumes the value of 0.1159 with a standard deviation of 0.0332. The histogram of the inconsistency rates provides a more comprehensive view at the results: there is a visible presence of a longer tail of the distribution spread towards higher values of the inconsistency index.

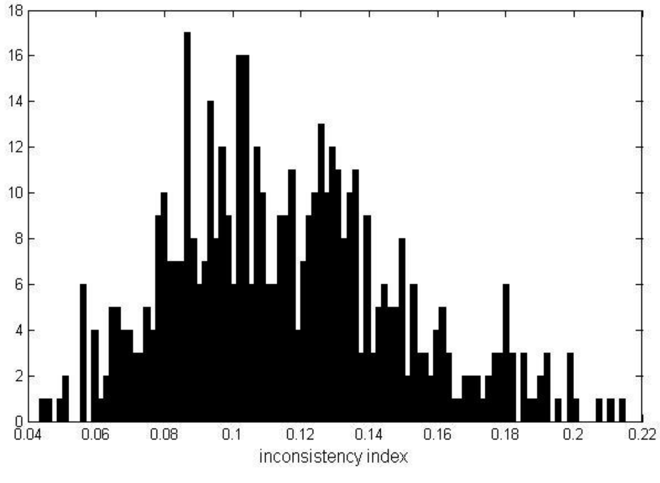


Figure 89. The distribution of the values of the inconsistency index- a uniform distribution of the cutoff points

The PSO returns the optimal cutoff points of 2.2 and 2.4, which are evidently shifted towards the lower end of the scale. The inconstancy index takes on now lower values and is equal to 0.054 with the standard deviation of 0.0252. The corresponding histogram is shown in Figure 90.

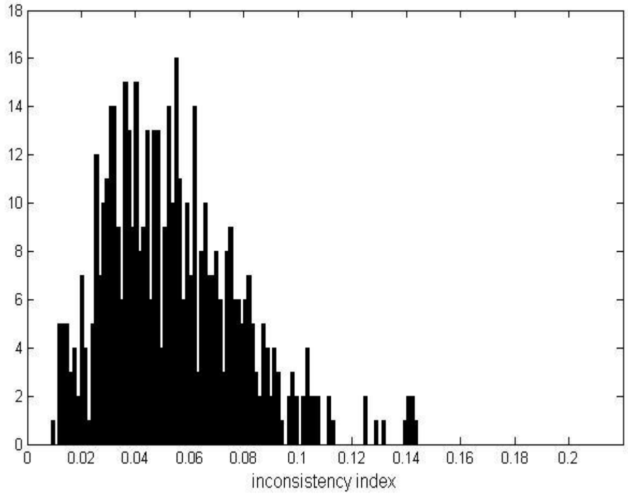


Figure 90. The distribution of the values of the inconsistency index- a PSO-optimized distribution of the cutoff points

For the optimal splits of the scale, a reciprocal matrix with the lowest inconsistency index is given below:

$$R = \begin{bmatrix} 1 & 2.2 & 1/1.65 & 1/2.35 & 1/2.4 \\ 1/2.2 & 1 & 1/2.1 & 1/4.05 & 1/6.15 \\ 1.65 & 2.1 & 1 & 1/1.45 & 1/2.45 \\ 2.35 & 4.05 & 1.45 & 1 & 1/1.25 \\ 2.4 & 6.15 & 2.45 & 1.25 & 1 \end{bmatrix}$$

with the normalized eigenvector corresponding to the largest eigenvalue of this reciprocal matrix equal to **e** = [0.79 1.00 0.65 0.30 0.00]^T, which identifies the second alternative as an optimal one.

Example 2 Here we consider another 5x5 reciprocal matrix with 5 linguistic terms, VL, L, M, H, and VH

$$R = \begin{bmatrix} 1 & L & M & VL & H \\ 1/L & 1 & M & 1/VL & VH \\ 1/M & 1/M & 1 & 1/H & L \\ 1/VL & VL & H & 1 & H \\ 1/H & 1/VH & 1/L & 1/H & 1 \end{bmatrix}$$

The results of the optimization are shown in Figure 91, here the parameters of the PSO were set up as follows: number of particles: 100, number of iterations: 500.

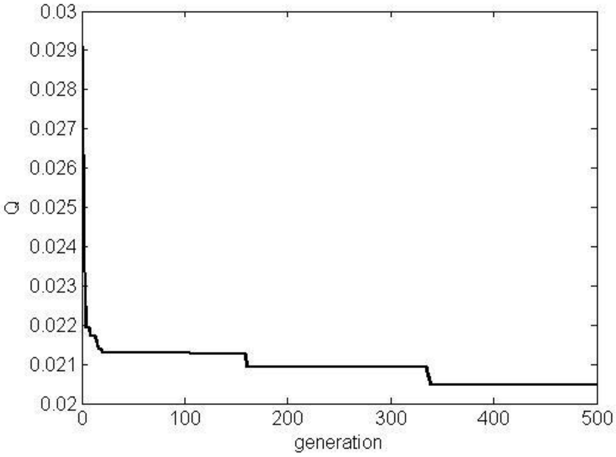
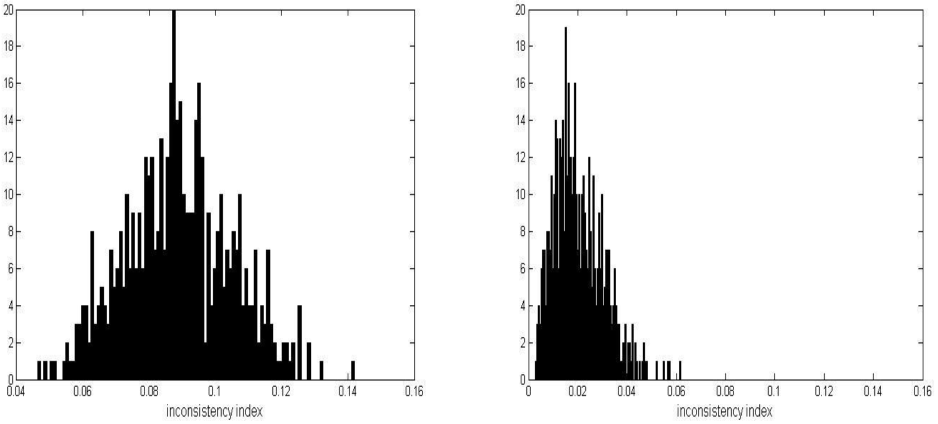


Figure 91. The values of the fitness function obtained in successive generations of the PSO

The results corresponding with the uniform distribution of the cutoff points (that is 2.6, 4.2, 5.8, and 7.4) come with the average inconsistency index of 0.0888 with a standard deviation of 0.0162. The PSO produces the cutoff points of 1.1, 1.8, 5.3, and 5.6. The inconsistency index is now lower and equal to 0.0205 with the standard deviation of 0.0102. The corresponding histograms both for the uniform and PSO-optimized cutoff points are shown in Figure 92.



(a) (b)

Figure 92. The distribution of the inconsistency index: (a) uniform distribution of cutoff points, (b) PSO-optimized distribution of the cutoff points

For the optimal split of the scale, a reciprocal matrix with the lowest inconsistency index is given below

$$R = \begin{bmatrix} 1 & 1.1 & 4.55 & 1.05 & 5.5 \\ 1/1.1 & 1 & 3.85 & 1/1.05 & 6.05 \\ 1/4.55 & 1/3.85 & 1 & 1/5.3 & 1.3 \\ 1/1.05 & 1.05 & 5.3 & 1 & 5.3 \\ 1/5.5 & 1/6.05 & 1/1.3 & 1/5.3 & 1 \end{bmatrix}$$

and the corresponding eigenvalue is equal to **e** = [1.00 0.92 0.06 0.97 0.00]^T.

Example 3 We consider a 7x7 reciprocal matrix with 5 linguistic terms as before, that is VL, L, M, H, and VH

$$R = \begin{bmatrix} 1 & 1/M & 1/VH & 1/H & 1/M & 1/VL & 1/L \\ M & 1 & 1/L & 1/VL & 1/VL & M & M \\ VH & L & 1 & H & VL & H & VH \\ H & VL & 1/H & 1 & 1/L & L & L \\ M & VL & 1/VL & L & 1 & L & H \\ VL & 1/M & 1/H & 1/L & 1/L & 1 & L \\ L & 1/M & 1/VH & 1/L & 1/H & 1/L & 1 \end{bmatrix}$$

The parameters of the PSO are the same as in the previous examples. The PSO optimization takes place in the first generations of the method, Figure 93.

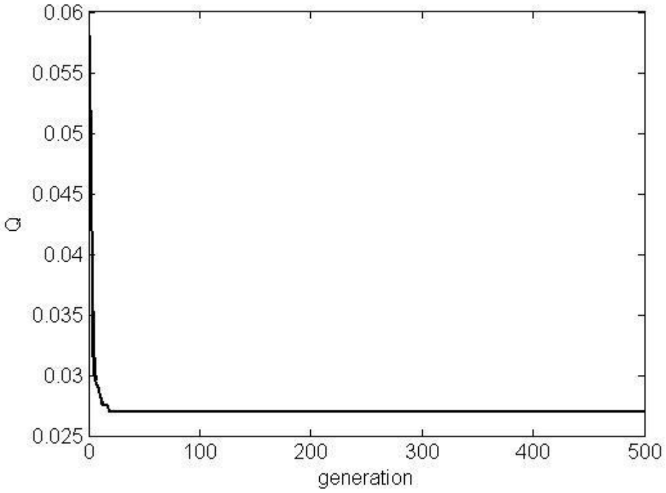


Figure 93. The values of the fitness function versus successive generations of the PSO

The obtained results are as follows:

Uniform distribution of the cutoff points: {2.6, 4.2, 5.8, 7.4}. The average inconsistency index assumes the value of 0.1184 with a standard deviation of 0.0153, the histogram of the inconsistency index is included in Figure 94.

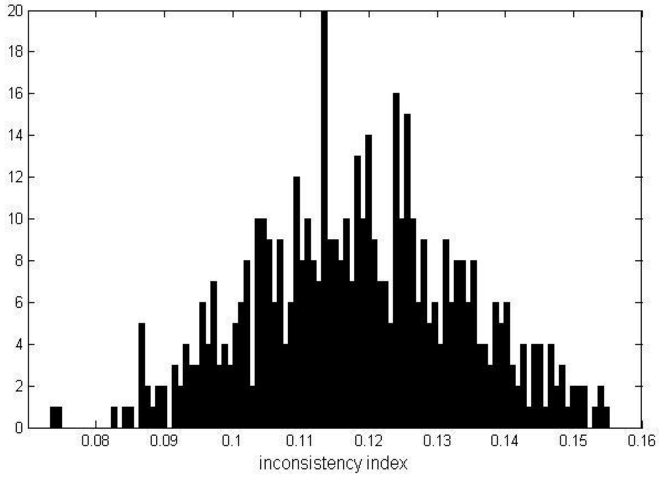


Figure 94. A histogram of distribution of values assumed by the inconsistency index – uniformly distributed cutoff points

PSO-optimized cutoff points: {1.5, 1.7, 2.3, 2.4}, average inconstancy index 0.0269 ±0.0075. The corresponding histogram of the values of the inconsistency index is illustrated in Figure 95. Not only the average value of the inconsistency index is lower but its standard deviation is also reduced to 50% of the one encountered when having a uniform distribution of the cutoff points.

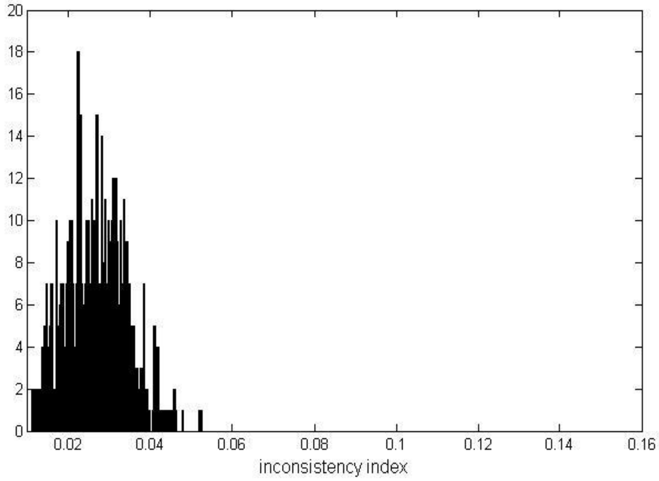


Figure 95. A histogram of distribution of values assumed by the inconsistency index – PSO optimized cutoff points

For the optimal splits of the [1, 9] interval, a reciprocal matrix with the lowest inconsistency index is given below

$$R = \begin{bmatrix} 1 & 1/2.15 & 1/3.9 & 1/2.3 & 1/2.25 & 1/1.25 & 1/1.6 \\ 2.15 & 1 & 1/1.65 & 1 & 1/1.05 & 1.8 & 2.2 \\ 3.9 & 1.65 & 1 & 2.3 & 1.15 & 2.35 & 3.8 \\ 2.3 & 1 & 1/2.3 & 1 & 1/1.5 & 1.65 & 1.55 \\ 2.25 & 1.05 & 1/1.15 & 1.5 & 1 & 1.6 & 2.35 \\ 1.25 & 1/1.8 & 1/2.35 & 1/1.65 & 1/1.6 & 1 & 1.55 \\ 1.6 & 1/2.2 & 1/3.8 & 1/1.55 & 1/2.35 & 1/1.55 & 1 \end{bmatrix}$$

the eigenvector associated with the maximal eigenvalue is **e** = [0 0.47 1.00 0.36 0.59 0.16 0.06]^T. For comparison, the uniform distribution of the cutoff points yields **e** = [1.00 0.63 0.00 0.73 0.43 0.91 0.97]^T.

10.3. Fuzzy Sets in the Quantification of Linguistic Terms

The quantification of the linguistic terms with the aid of fuzzy sets can be realized by further refining the representation completed through intervals of the scale. The intent is to form fuzzy sets by using the cutoff points already formed when building the intervals. This helps assess the performance of this model of the linguistic terms versus the one of the interval character.

Consider the intervals already formed over the [1, 9] scale with the cutoff points shown in Figure 96.

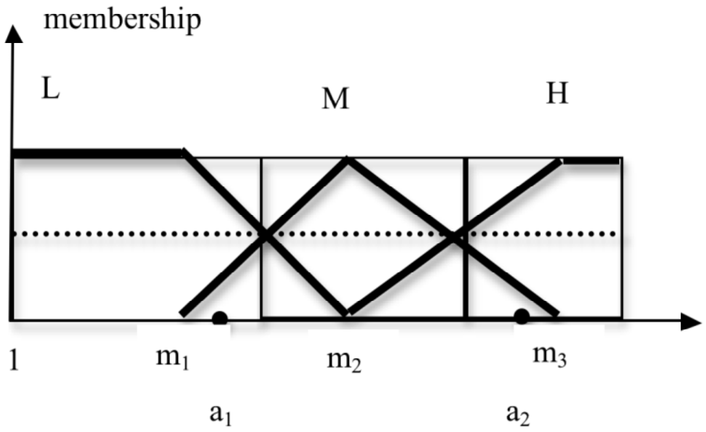


Figure 96. Formation of fuzzy sets of quantification of the linguistic terms based on the already constructed interval-valued representations of the terms, shown are calculations of the activation levels of the terms L and H for randomly drawn sample with the values a_1 and a_2.

Based on them we form triangular fuzzy sets whose modal values are adjustable. In the construct above the fuzzy sets overlap at 0.5 at the cutoff points as shown in Figure 96.

When selecting randomly the realization of the linguistic terms, they come with the membership values computed on the basis of the membership functions. For instance, if we randomly draw the values a_1 and a_2 as the realizations of the linguistic terms Low (Low) and High (H) present in the reciprocal matrix the results (viz. the eigenvalue and its associated eigenvector) come with the figure of merit computed based on the aggregation of the membership degrees of L and H that is $L(a_1)$ t $H(a_2)$ where "t" stands for any t-norm. Note that the values are drawn randomly from the supports of the linguistic terms L and H. Denote it by μ where $\mu = L(a_1)$ t $H(a_2)$. The calculated inconsistency index ν is now associated with the value of μ. In a nutshell, we slightly augment the procedure of evaluating the fitness function discussed for the interval realization of linguistic terms. The main difference is that in the calculations of the fitness function used by the PSO to optimize the modal values of the fuzzy sets of the linguistic terms we use the weighted sum of the form

$$\frac{\sum_{k=1}^{K} \mu_k \nu_k}{\sum_{k=1}^{K} \mu_k} \tag{10.1}$$

with the summation completed over all the samples drawn, k=1,2, …, K. The above formula reduces to the one used in the previous case when the membership functions are replaced by the characteristic functions.

An alternative approach is to start with the construction of fuzzy sets of the linguistic terms. In this case the content of the particle is a collection of the modal values of the triangular membership functions. We assume that the consecutive membership functions intersect at the level of 1/2; this implies that the fuzzy sets are completely specified by their modal values.

Example 4 We use the same reciprocal matrix as in Example 1. Using interval granulation, the best cutoffs of interval are 2.2 and 2.4. By making use of the fuzzy sets in scale granulation and with the sample of 500 elements and the fitness function expressed by (2.27) the average fitness function Q is equal to 0.0533, and other results are from 0.0533 to 0.0635. The PSO optimized fuzzy sets are shown in Figure 97.

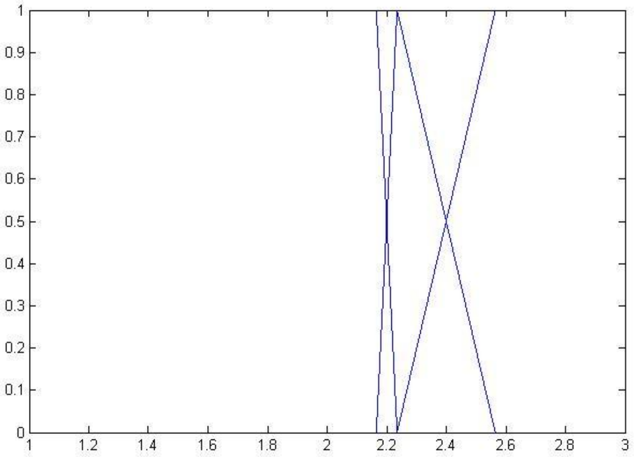


Figure 97. Optimized membership functions of triangular fuzzy sets

The fuzzy set of preferences comes as **e** = [0.84 1.00 0.83 0.47 0.00]^T.

Proceeding with the second approach, viz. constructing fuzzy sets from scratch, the PSO-optimized cutoff points are 1.4, 2.6, and 2.8 for which the performance index Q is equal to 0.051 (which is very close to the value of the fitness function obtained in the first approach). The progression of the optimization process is illustrated in Figure 98.

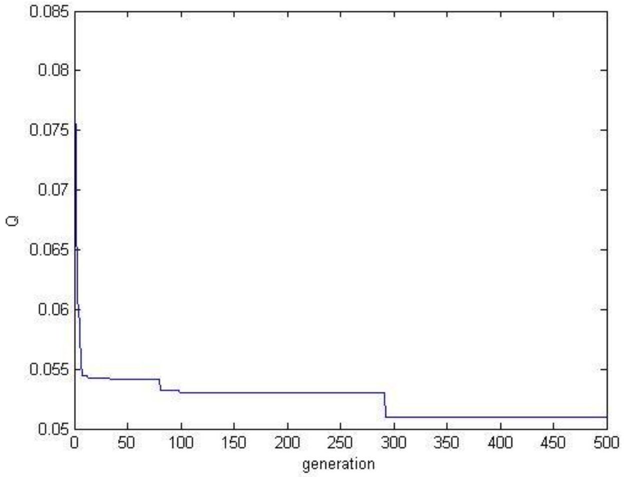


Figure 98. Q in successive PSO generations

The fuzzy set of preferences has the following entries **e** = [0.81 1.00 0.84 0.37 0.00]^T. The results are practically the same as obtained when starting with the interval quantification of the linguistic terms.

10.4. A Group Decision-Making Scenario

The AHP model is used in the realization of group decision-making processes. Here we assume “c” reciprocal matrices. Each decision-maker uses the same number of linguistic terms. The optimization is realized in the same way as before. The main difference is with regard to the performance index which is taken as a sum of the inconsistency indexes for all reciprocal matrices R[1], R[2], … R[c]. Here we distinguish between two scenarios, namely (a) each reciprocal matrix involves the same number of linguistic terms, and (b) there are different numbers of the linguistic terms. The fitness function to be minimized is the sum of the inconsistency indexes of the corresponding reciprocal matrices, namely

$$Q = \sum_{i=1}^{c} \nu[i] \tag{10.2}$$

The joint treatment of the linguistic terms coming from the experts engaged in the process of group decision-making allows us to treat these terms in a unified fashion and reconcile their semantics so that the individual rankings are made comparable and thus could be aggregated to arrive at the joint view at the alternatives. A granulation of the linguistic terms realized at the level of individual decision-makers involved in the group decision-making may result in results of individual rankings that are more difficult to aggregate.

Example 5. We consider 4 decision-makers whose assessments of alternatives are presented in the form of the following reciprocal matrices (with 5 linguistic terms); note there is the same number of linguistic terms used in all reciprocal matrices

$$R_1 = \begin{bmatrix} 1 & L & M & VL & H \\ 1/L & 1 & M & 1/VL & VH \\ 1/M & 1/M & 1 & 1/H & L \\ 1/VL & VL & H & 1 & H \\ 1/H & 1/VH & 1/L & 1/H & 1 \end{bmatrix} \quad R_2 = \begin{bmatrix} 1 & VL & 1/L & 1/M & 1/VH \\ 1/VL & 1 & 1/L & 1/H & 1/H \\ L & L & 1 & 1/L & 1/VH \\ M & H & L & 1 & 1/VL \\ VH & H & VH & VL & 1 \end{bmatrix}$$

$$R_3 = \begin{bmatrix} 1 & H & VH & M & VH \\ 1/H & 1 & 1/L & 1/M & VL \\ 1/VH & L & 1 & 1/VL & L \\ 1/M & M & VL & 1 & H \\ 1/VH & 1/VL & 1/L & 1/H & 1 \end{bmatrix} \quad R_4 = \begin{bmatrix} 1 & 1/L & 1/VL & 1/VH & 1/M \\ L & 1 & 1/L & 1/VH & 1/L \\ VL & L & 1 & 1/M & 1/VL \\ VH & VH & M & 1 & H \\ M & L & VL & 1/H & 1 \end{bmatrix}$$

One should note that the linguistic terms L, M, etc. even used by all decision-makers may have slightly different meaning resulting in different information granules. The parameters of the PSO were set up as follows: the number of particles is set to 100 (the larger population was tried as well however the result is similar), number of iterations is equal to 500. The results in terms of the average inconsistency indexes of the reciprocal matrices are included in Figure 99.

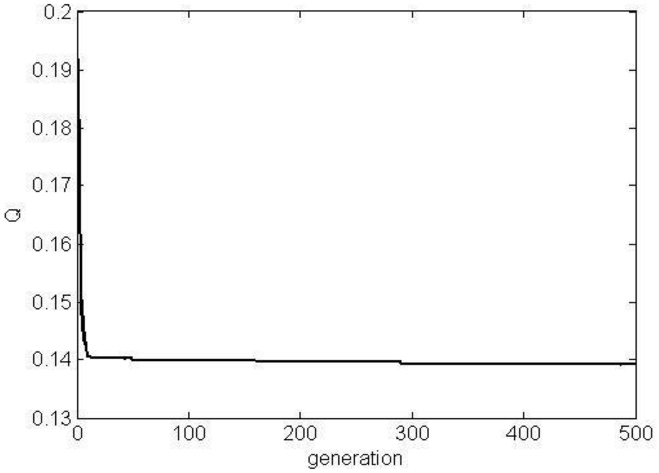


Figure 99. Q in successive PSO generations

The minimized fitness function is now equal to 0.1392 ± 0.049. The corresponding preference vectors associated with the maximal eigenvalues of the reciprocal matrices are as follows

e[1] = [1.00 0.69 0.12 0.97 0.00]^T
e[2] = [1.00 0.99 0.83 0.34 0.00]^T
e[3] = [1.00 0.07 0.15 0.37 0.00]^T
e[4] = [1.00 0.95 0.87 0.00 0.85]^T

The cutoff points obtained here are 1.5, 1.8, 3.8, and 4.0. The realization of the linguistic terms was unified throughout the group of experts. In contrast, let us consider the same reciprocal matrices and come up with the granulation of the scale done individually. This means that four separate optimization problems are solved. The results are as follows:

R$_1$: Q = 0.0013, **e** = [1.00 0.95 0.03 0.99 0]^T, cutoff points: 1.1, 1.8, 5, 5.1
R$_2$: Q = 0.0174, **e** = [0.004 0 0.11 0.53 1.00]^T, cutoff points: 1.8, 2.6, 6.7,6.8
R$_3$: Q = 0.0071, **e** = [0 0.95 0.85 0.65 1.00]^T, cutoff points:1.6, 1.7, 3.2, 4.7
R$_4$: Q = 0.0058, **e** = [1.00 0.93 0.84 0 0.79]^T, cutoff points:1.5, 1.7, 3, 3.1

The comparison of performance of uniform distribution of cutoff points and optimal distribution is shown in Figures 100 and 101.

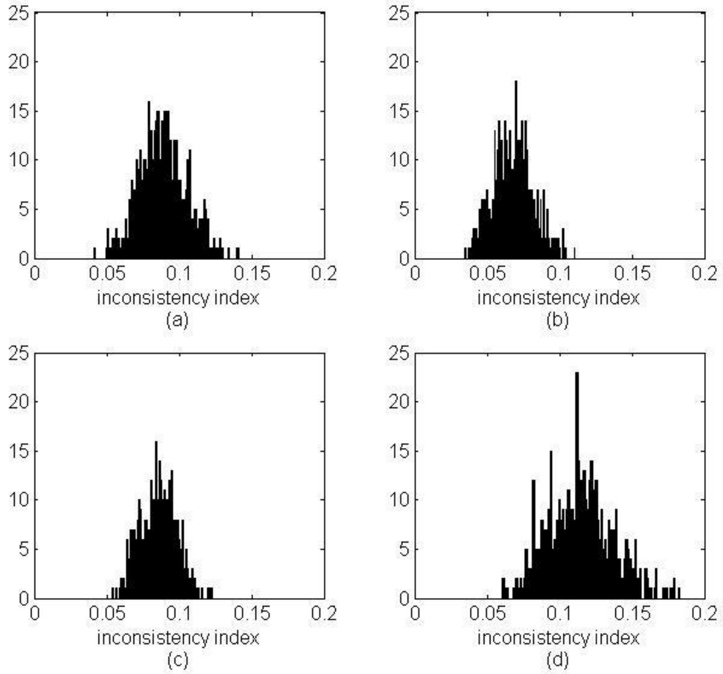


Figure 100. The distribution of the inconsistency index in case of a uniform distribution of cutoff points: (a) R_1 (b) R_2 (c) R_3 (d) R_4

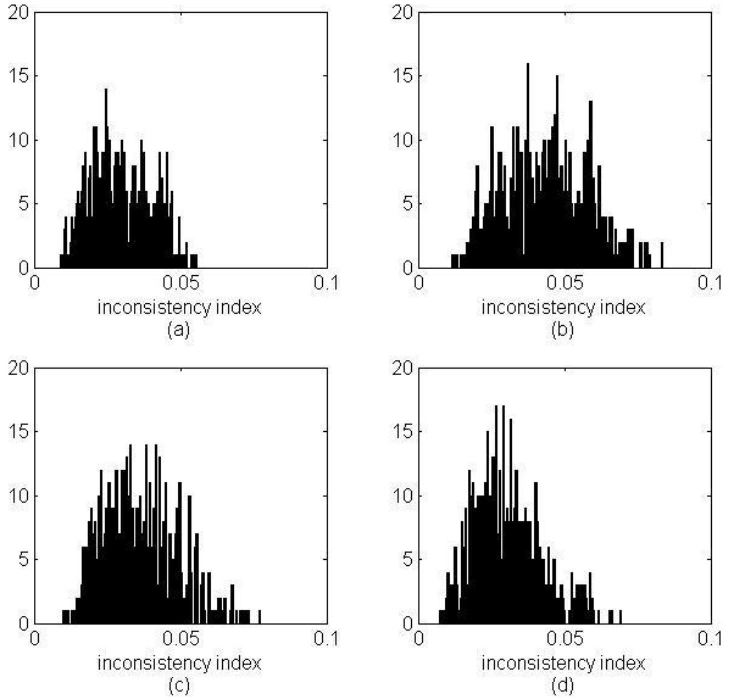


Figure 101. The PSO-optimized distribution of the inconsistency index: (a). R_1 (b), R_2 (c) R_3 (d) R_4

In the previous example, the reciprocal matrices have the same number of linguistic terms. In the next example, we consider different numbers of linguistic terms present in the reciprocal matrices. This requires that some mapping between the linguistic terms is defined so that the corresponding information granules can be constructed.

Example 6 Here we use several reciprocal matrices each of them exhibiting a different number of linguistic entries. More specifically, we have the following collections of linguistic terms

R_1 - {L_1, M_1, H_1}, R_2 - {VL_2, L_2, M_2, H_2, VH_2}, R_3- {L_3, M_3, H_3}, R_4 – {VL_4, L_4, M_4, H_4}

We consider the following association between the linguistic terms present in the individual reciprocal matrices or form new aggregated (concatenated) linguistic terms L, M, H, namely

$$
\begin{aligned}
&L = \{VL_2, L_2\} = L_3 = \{VL_4, L_4\},\\
&M = M_1=M_2=M_3=M_4,\\
&H = \{H_2, VH_2\} = H_3 =H_4.
\end{aligned}
$$

The optimization of the cutoff points (whose number is equal to 4) is completed for the newly defined linguistic terms with the interval-valued granulation. We consider 4 decision-makers whose assessments of alternatives are presented in the form of the following reciprocal matrices (all are 5-dimension); note there is the different numbers of linguistic terms used in each reciprocal matrix.

$$
R_1 = \begin{bmatrix} 1 & L_1 & M_1 & M_1 & H_1 \\ 1/L_1 & 1 & M_1 & H_1 & M_1 \\ 1/M_1 & 1/M_1 & 1 & 1/L_1 & L_1 \\ 1/M_1 & 1/H_1 & L_1 & 1 & L_1 \\ 1/H_1 & 1/M_1 & 1/L_1 & 1/L_1 & 1 \end{bmatrix}
\qquad
R_2 = \begin{bmatrix} 1 & 1/L_2 & 1/H_2 & 1/VL_2 & 1/M_2 \\ L_2 & 1 & 1/VL_2 & 1/VL_2 & 1/VH_2 \\ H_2 & VL_2 & 1 & M_2 & 1/VL_2 \\ VL_2 & VL_2 & 1/M_2 & 1 & 1/M_2 \\ M_2 & VH_2 & VL_2 & M_2 & 1 \end{bmatrix}
$$

$$
R_3 = \begin{bmatrix} 1 & M_3 & H_3 & L_3 & M_3 \\ 1/M_3 & 1 & M_3 & L_3 & L_3 \\ 1/H_3 & 1/M_3 & 1 & 1/M_3 & L_3 \\ 1/L_3 & 1/L_3 & M_3 & 1 & H_3 \\ 1/M_3 & 1/L_3 & 1/L_3 & 1/H_3 & 1 \end{bmatrix}
\qquad
R_4 = \begin{bmatrix} 1 & VL_4 & H_4 & 1/VL_4 & M_4 \\ 1/VL_4 & 1 & VL_4 & VL_4 & M_4 \\ 1/H_4 & 1/VL_4 & 1 & 1/L_4 & L_4 \\ VL_4 & 1/VL_4 & L_4 & 1 & H_4 \\ 1/M_4 & 1/M_4 & 1/L_4 & 1/H_4 & 1 \end{bmatrix}
$$

The parameters of the PSO were set up as follows: the number of particles is set to 100 (the larger population was tried as well however the result is similar), and number of iterations is equal to 500. The minimized fitness function is now equal to 1.1289 ± 0.1045. The corresponding preference vectors associated with the maximal eigenvalues of the reciprocal matrices are as follows

e[1] = [1.00 0.99 0.98 0.00 0.24]^T
e[2] = [1.00 0.98 0.11 0.09 0.00]^T
e[3] = [0.00 0.01 0.02 1.00 0.76]^T
e[4] = [0.00 0.52 0.56 0.65 1.00]^T

The cutoff points obtained here are 1.7, 3.3, 3.4, 6.3. If we solve the optimization problem separately for each matrix, the results are:

R_1: Q = 0.0549, **e** = [0 0.10 0.98 0.99 1.00], cutoff points: 1.6, 8.1
R_2: Q = 0.0601, **e** = [1.00 0.79 0.13 0.82 0], cutoff points: 1.6, 1.7, 2.1, 2.3
R_3: Q = 0.0931, **e** = [1.00 0.42 0.04 0.57 0], cutoff points: 1.6, 2.9
R_4: Q = 0.0809, **e** = [0.98 0.69 0.22 1.00 0], cutoff points: 1.4, 3.6, 3.7

10.5. Conclusions

The study presented here provided the methodology and the algorithmic framework of constructing preference vectors on a basis of reciprocal matrices with the linguistically quantified results of pairwise comparisons. It is important to underline that while linguistic information is readily available, it is not operational and thus the phase of information granulation becomes indispensible. The mapping of the linguistic assessments to the corresponding information granules makes the linguistic information operational so that the final preference values are determined. The formation of a suite of information granules realizing a linguistic (symbolic) quantification as a result of a given optimization problem equips these granules with well-articulated semantics.

The use of PSO as an optimization environment offers a great deal of flexibility. Different fitness functions could be easily accommodated. Likewise a multiobjective optimization can be sought. The need for the two-objective optimization becomes apparent in case of a group decision-making where in addition to the criterion of consistency of evaluations applied to the individual reciprocal matrices, one can consider a minimization of dispersions among the vectors of preferences associated with these matrices.

While we have predominantly focused our discussion on the granulation of linguistic terms in the language of intervals and fuzzy sets (to some extent), the discussed methodology applies equally well in case of other formalisms of information granules. In particular, dealing with probabilistically granulated linguistic terms could help shed light on possible linkages between probabilistic and fuzzy models of decision-making along with some possible hybrid probabilistic-fuzzy schemes.

11. Conclusions

In this research, a new system modeling framework with granular architectures was introduced, developed and experimented with. It is very difficult to analyze a system or phenomenon with the use of a single monolithic (and quite commonly complex) model. In this study, a complex system modeling problem is addressed by firstly constructing some local models, which are then aggregated through some mechanisms of collaboration and resulting in higher level granular models. The term “granular” reflects upon and quantifies the diversity of the local models. Knowledge collaboration and reconciliation is used to establish modeling outcomes of global nature. The diversity of views at the system/problem/phenomenon is quantified via the granularity of results produced by the global model constructed at the higher level.

In the development process, several essential problems have been solved aiming at achieving fullest usage of knowledge at hand and better performance of the ultimate model in presence of new data. Firstly, a reduction of feature space is realized by real-coded genetic algorithm prior to the construction of local models. The sources of knowledge (local models) are quantified by information granularity. It is also shown the construction of information granules completed through the use of the principle of justifiable granularity which deals with two conflicting requirements.

The approach itself is rational as information granularity can provide some flexibility to quantify the diversity of available sources of knowledge, optimize parameters of a model, and adjust decisions of decision makers. We have demonstrated the effectiveness of the techniques used to generate information granules by synthetic data in order to better understand the benefits of information granulation and its ability. Our approach is then successfully applied through several different applications including

- granular fuzzy systems
- granular neural networks
- granular analytic hierarchy process

For example, with a granular fuzzy system, when dealing with a collection of individual models, aggregation methods studied in the literature produce the result at the same level of granularity as the results generated by individual models involved in the aggregation. This strategy fails to reflect upon and quantify the diversity of the available individual models, shows poor ability in evaluating the quality of the results of modeling. In this way we are able to demonstrate the advantages of using information granularity as a design asset. We have shown the most essential features of the proposed methodology and algorithmic development through numeric experiments and obtained results.

The main contributions of our study are

- providing a mechanism for forming meaningful representation of numeric data
- optimizing the distribution of information granularity by PSO
- providing a way to operate on linguistic information
- reducing the input feature space of data
- reconciling the knowledge coming from local available models
- producing a granular outcome of a model
- generating interval representations of reciprocal matrices
- solving two-objective optimization by PSO

The theoretical development of system modeling with granular architecture is still very much incomplete since a number of open research problems remain including

- realizing modeling with other formalisms of information granules
- choosing a most representative level of granularity
- comparing PSO results with other optimization methods
- decreasing complexity of the algorithm and the computation time

The granular representation of neural networks discussed above was the one using intervals; however, any other formalisms of granular computing, especially fuzzy sets, could be equally applicable here. In this way, all weights and/or biases are extended into fuzzy sets and the output of the granular neural network is in the fuzzy set formalism.

We envision from above discussion that while granular models can offer substantial advantages over the existing “conventional” models, it opens up interesting new applications in the future that could otherwise not be contemplated.

Bibliography

Aguilar-Lasserre, A.A., M.A. Bautista Bautista, A. Ponsich, and M.A. Gonzalez Huerta. "An AHP-based decision-making tool for the solution of multiproduct batch plant design problem under imprecise demand." *Computers and Operations Research* 36, no. 3 (2009): 711-736.

Akhand, M.A.H., Md. Monirul Islam, and K. Murase. "Progressive interactive training: A sequential neural network ensemble learning method." *Neurocomputing* 73, no. 1-3 (2009): 260-273.

Alcala, R., P. Ducange, F. Herrera, B. Lazzerini, and F. Marcelloni. "A multiobjective evolutionary approach to concurrently learn rule and data bases of linguistic fuzzy-rule-based systems." *IEEE Trans on Fuzzy Systems* 17, no. 5 (2009): 1106-1122.

Alonso, S., E. Herrera-Viedma, F. Chiclana, and F. Herrera. "A web based consensus support system for group decision making problems and incomplete preferences." *Information Sciences* 180, no. 23 (2010): 4477-4495.

Altuzarra, A., J.M. Moreno-Jimenez, and M. Salvador. "A Bayesian priorization procedure for AHP-group decision making." *European Journal of Operational Research* 182, no. 1 (2007): 367-382.

Apostolou, B., and J.M. Hassell. "Note on consistency ratio: a reply." *Mathematical and Computer Modeling* 35, no. 9-10 (2002): 1081-1083.

Araujo, E., and L.S. Coelho. "Particle swarm approaches using Lozi map chaotic sequences to fuzzy modeling of an experimental thermal-vacuum system." *Applied Soft Computing* 8, no. 4 (2008): 1354-1364.

Auda, G., and M. Kamel. "CMNN: Cooperative modular neural networks." *Neurocomputing* 20, no. 1-3 (1998): 189-207.

Bargiela, A., and W. Pedrycz. *Granular Computing: An introduction.* Kluwer Academic, 2003.

Bergh, F., and A.P. Engelbrecht. "A study of particle swarm optimization particle trajectories." *Information Sciences* 176, no. 8 (2006): 937-971.

Beynon, M.J. "Understanding local ignorance and non-specificity within the DS/AHP method of multi-criteria decision making." *European Journal of Operational Research* 163, no. 2 (2005): 403-417.

Bezdek, J.C. *Pattern Recognition with Fuzzy Objective Function Algorithms.* New York: Plenum Press, 1981.

Black, M. "Vagueness: an exercise in logical analysis." *Philosophy of Science* 4 (1937): 427-455.

Bloomfield, P. *Fourier analysis of time series: An introduction.* New York: Wiley, 1976.

Blum, E.K., and L.K. Li. "Approximation theory and feedforward." *Neural Networks* 4 (1991): 511-515.

Bonissone, P.P., A. Varma, K.S. Aggour, and F. Xue. "Design of local fuzzy models using evolutionary algorithms." *Computational Statistics and Data Analysis* 51 (2006): 398-416.

Bordogna, G., M. Fedrizzi, and G. Pasi. "A linguistic modeling of consensus in group decision making based on OWA operators." *IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans* 27, no. 1 (1997): 126-133.

Bortolan, G. "An architecture of fuzzy neural networks for linguistic processing." *Fuzzy Sets and Systems* 100 (1998): 197-215.

Brillinger, D.R. *Time series: Data analysis and theory.* New York: Holt, Rinehart and Winston, 1975.

Buckley, J., and Y. Hayashi. "Fuzzy neural networks: a survey." *Fuzzy Sets and Systems* 66 (1994): 1-14.

Cakir, O. "On the order of the preference intensities in fuzzy AHP." *Computers and Industrial Engineering* 54, no. 4 (2008): 993-1005.

Chang, T.H., and T.C. Wang. "Using the fuzzy multi-criteria decision making approach for measuring the possibility of successful knowledge management." *Information Sciences* 9, no. 4 (2009): 355-370.

Chen, H., P. Tiho, and X. Yao. "Predictive Ensemble Pruning by Expectation Propagation." *IEEE Trans on Knowledge and Data Engineering* 21, no. 7 (2009): 999-1013.

Chen, J., and Y. Xi. "Nonlinear system modeling by competitive learning and adaptive fuzzy inference system." *IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 28, no. 2 (1998): 231-238.

Chen, L., C.L.P. Chen, and W. Pedrycz. "A gradient-descent-based approach for transparent linguistic interface generation in fuzzy models." *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics* 40, no. 5 (2010): 1219-1230.

Chen, W.H., and W.X. Zheng. "Robust stability analysis for stochastic neural networks with time-varying delay." *IEEE Trans on Neural Networks* 2, no. 3 (2010): 508-514.

Chu, P., and J.K.H. Liu. "Note on consistency ratio." *Mathematical and Computer Modeling* 35, no. 9-10 (2002): 1077-1080.

Clerc, M., and J. Kennedy. "The particle swarm–explosion, stability, and convergence in a multidimensional complex space." *IEEE Trans on Evol. Comput.* 6, no. 1 (2002): 58-73.

Coelho, L.S., L.Z. Barbosa, and L. Lebensztajn. "Multiobjective particle swarm approach for the design of a brushless DC wheel motor." *IEEE Trans on Magnetics* 46, no. 8 (2010): 2994-2997.

Coello Coello, C.A., and M.S. Lechuga. "MOPSO: A proposal for multiple objective particle swarm optimization." *Proc. Congr. Evol. Comput.* Honolulu, 2002. 1051-1056.

Coello Coello, C.A., G.T. Pulido, and M.S. Lechuga. "Handling multiple objectives with particle swarm optimization." *IEEE Trans on Evolutionary Computation* 8, no. 3 (2004): 256-279.

Cybenco, G. "Approximation by superpositions of a sigmoidal function." *Math. Contr., Signals Syst.* 2 (1989): 303-314.

Das, R., I. Turkoglu, and A. Sengur. "Effective diagnosis of heart disease through neural networks ensembles." *Expert Systems with Applications* 36, no. 4 (2009): 7675-7680.

David Sánchez, A.V. "Searching for a solution to the automatic RBF network design problem." *Neurocomputing* 42, no. 1-4 (2002): 147-170.

de Weerdt, E., Q.P. Chu, and J.A. Mulder. "Neural network output optimization using interval analysis." *IEEE Trans on Neural Networks* 20, no. 4 (2009): 638-653.

Delgado, M., F. Gomez-Skarmeta, and F. Martin. "A fuzzy clustering based rapid prototyping for fuzzy rule-based modeling." *IEEE Trans on Fuzzy Systems* 5, no. 2 (1997): 223-233.

DeSanctis, G., and R.B. Gallupe. "A foundation for the study of group support systems." *Manage Science* 33, no. 5 (1987): 589-609.

Dick, S., A. Schenker, W. Pedrycz, and A. Kandel. "Regranulation: A granular algorithm enabling communication between granular worlds." *Information sciences* 177, no. 2 (2007): 408-435.

DiNola, A., S. Sessa, W. Pedrycz, and E. Sanchez. *Fuzzy relational equations and their applications in knowledge engineering.* Dordrecht: Kluwer Academic Publishers, 1989.

Dong, Y., G. Zhang, W-C. Hong, and Y. Xu. "Consensus models for AHP group decision making under row geometric mean prioritization method." *Decision Support Systems* 49, no. 3 (2010): 281-289.

Dong, Y., W.C. Hong, Y. Xu, and S. Yu. "Selecting the individual numerical scale and prioritization method in the analytic hierarchy process: A 2-tuple fuzzy linguistic approach." *IEEE Trans on Fuzzy Systems* 19, no. 1 (2011): 13-25.

Dorigo, M., V. Maniezzo, and A. Colorni. "The ant system: Optimization by a colony of cooperating agents." *IEEE Trans on System, Man, and Cybernetics: Part B: Cybernetics* 26, no. 1 (1996): 29-41.

Du, D., K. Li, and M. Fei. "A fast multi-output RBF neural network construction method." *Neurocomputing* 73, no. 10-12 (2010): 2196-2202.

Dubois, D., and H. Prade. "The three semantics of fuzzy sets." *Fuzzy Sets and Systems* 90 (1997): 141-150.

Duda, R.O., P.E. Hart, and D.G. Stork. *Pattern Classification.* New York: Wiley, 2001.

Duhem, P. "The aim and structure of physical theory." Paris, 1906.

Eftekhari, M., S.D. Katebi, M. Karimi, and A.H. Jahanmiri. "Eliciting transparent fuzzy model using differential evolution." *Applied Soft Computing* 8, no. 1 (2008): 466-476.

Fang, R., J. Zhou, F. Liu, and B. Peng. "Short-term load forecasting using interval arithmetic backpropagation neural network." *International Conference on Machine Learning and Cybernetics.* 2006. 2872-2876.

Farag, W.A., V.H. Quintana, and G. Lambert-Torres. "A genetic-based neuro-fuzzy approach for modeling and control of dynamical systems." *IEEE Trans on Neural Networks* 9, no. 5 (1998): 756-767.

Fishman, G.S. *Monte Carlo: Concepts, Algorithms, and Applications.* New York: Springer, 1995.

Fonseca, C.M., and P.J. Fleming. "Multiobjective optimization and multiple constraint handling with evolutionary algorithms–Part I: A unified formulation." *IEEE Trans on Systems, Man and Cybernetics, Part A: Systems and Humans* 28, no. 1 (1998): 26-37.

Frei, F.X., and P.T. Harker. "Measuring aggregate process performance using AHP." *European Journal of Operational Research* 116, no. 2 (1999): 436-442.

Freund, Y., and R.E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of Computer and System Sciences* 55, no. 1 (1997): 119-139.

Fumera, G., and F. Roli. "A theoretical and experimental analysis of linear combiners for multiple classifier systems." *IEEE Trans on Pattern Analysis and Machine Intelligence* 27, no. 6 (2005): 942-956.

Funahashi, K.I. "On the approximate realization of continuous mappings." *Neural Networks* 2 (1989): 183-192.

Garcia-Lapresta, J.L. "A general class of simple majority decision rules based on linguistic opinions." *Information Sciences* 176, no. 4 (2006): 352-365.

Garczarczyk, Z.A. "Interval neural networks." *The 2000 IEEE International Symposium on Circuits and Systems.* 2000. 567-570.

Gobi, A.F., and W. Pedrycz. "Fuzzy modeling through logic optimization." *International Journal of Approximate Reasoning* 45, no. 3 (2007): 488-510.

Goldberg, D.E. *Genetic Algorithms in Search, Optimization, and Machine.* New York: Addison-Wesley, 1989.

Granitto, P.M., P.F. Verdes, and H.A. Ceccatto. "Neural network ensembles: evaluation of aggregation algorithms." *Artificial Intelligence* 163, no. 2 (2005): 139-162.

Green, M., U. Ekelund, L. Edenbrandt, J. Bjork, J.L. Forberg, and M. Ohlsson. "Exploring new possibilities for case-based explanation of artificial neural network ensembles." *Neural Networks* 22, no. 1 (2009): 75-81.

Herrera, F., E. Herrera-Viedma, and J.L. Verdegay. "A rational consensus model in group decision making using linguistic assessments." *Fuzzy Sets and Systems* 88, no. 1 (1997): 31-49.

Herrera, F., L. Martinez, and P.J. Sanchez. "Managing non-homogeneous information in group decision making." *European Journal of Operational Research* 166, no. 1 (2005): 115-132.

Herrera-Viedma, E., L. Martinez, F. Mata, and F. Chiclana. "A consensus support system model for group decision-making problems with multigranular linguistic preference relations." *IEEE Trans on Fuzzy Systems* 13, no. 5 (2005): 644-658.

Herrera-Viedma, E., S. Alonso, F. Chiclana, and F. Herrera. "A consensus model for group decision making with incomplete fuzzy preference relations." *IEEE Trans on Fuzzy Systems* 15, no. 5 (2007): 863-877.

Hirota, K., and W. Pedrycz. "Fuzzy relational compression." *IEEE Trans on Systems, Man and Cybernetics, Part B: Cybernetics* 29, no. 3 (1999): 407-415.

Hong, X., and S. Chen. "Modeling of complex-valued wiener systems using B-spline neural network." *IEEE Trans on Neural Networks* 22, no. 5 (2011): 818-825.

Hopfield, J.J. "Neural networks and physical systems with emergent collective computational abilities." *Proceedings of the National Academy of Sciences of the USA.* 1982. 2554-2558.

Horn, J. "Multiple decision making." In *Handbook of Evolutionary Computation*, by D.Fogel and Z.Michalewicz. London: Oxford Univ. Press, 1997.

Hornik, K., M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators." *Neural Networks* 2 (1989): 359-366.

Hsiao, W.F., H.H. Lin, and T.M. Chang. "Fuzzy consensus measure on verbal opinions." *Expert Systems with Applications* 35, no. 3 (2008): 836-842.

Hu, S., and J. Wang. "Global robust stability of a class of discrete-time interval neural networks." *IEEE Trans on Circuits and Systems I: Regular Papers* 53, no. 1 (2006): 129-138.

Hu, X., and R.C. Eberhart. "Multiobjective optimization using dynamic neighborhood particle swarm optimization." *Proc. Congr. Evol. Comput.* Honolulu, 2002. 1677-1681.

Huang, L., B. Zhang, and Q. Huang. "Robust interval regression analysis using neural networks." *Fuzzy Sets and Systems* 97, no. 3 (1998): 337-347.

Iatrou, M., T.W. Berger, and V.Z. Marmarelis. "Modeling of nonlinear nonstationary dynamic systems with a novel class of artificial neural networks." *IEEE Trans on Neural Networks* 10, no. 2 (1999): 327-339.

Ishibuchi, H., and H. Tanaka. "An architecture of neural networks with interval weights and its application to fuzzy regression analysis." *Fuzzy Sets and Systems* 57, no. 1 (1993): 27-39.

Ishibuchi, H., and M. Nii. "Improving the generalization ability of neural networks by interval arithmetic." *International Conference on Knowledge-based Intelligent Electronic Systems.* 1998. 231-236.

Ishibuchi, H., and Y. Nojima. "Analysis of interpretability-accuracy tradeoff of fuzzy systems by multiobjective fuzzy genetics-based machine learning." *International Journal of Approximate Reasoning* 44, no. 1 (2007): 4-31.

Ishibuchi, H., K. Kwon, and H. Tanaka. "A learning algorithm of fuzzy neural networks with triangular fuzzy weights." *Fuzzy Sets and Systems* 71, no. 3 (1995): 277-293.

Ishibuchi, Y. "Development of fuzzy neural networks." In *Fuzzy Modeling, Paradigms and Practice*, by W.Pedrycz, 185-202. Dordrecht: Kluwer Academic Publishers, 1996.

Islam, Md.M., X. Yao, S.M.S. Nirjon, M.A. Islam, and K. Murase. "Bagging and boosting negatively correlated neural networks." *IEEE Trans on Systems, Man, Cybernetics, Part B: Cybernetics* 38, no. 3 (2008): 771-784.

Janson, S., and M. Middendorf. "A hierarchical particle swarm optimizer and its adaptive variant." *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics* 35, no. 6 (2005): 1272-1282.

Jeonghwan, J., L. Rothrock, P.L. McDermott, and M. Barnes. "Using the analytic hierarchy process to examine judgement consistency in a complex multiattribute task." *IEEE Trans on Systems, Man and Cybernetics, Part A:Systems and Humans* 40, no. 5 (2010): 1105-1115.

Jin, Y. "Fuzzy modeling of high-dimensional systems: complexity reduction and interpretability improvement." *IEEE Trans on Fuzzy Systems* 8, no. 2 (2000): 212-221.

Juang, C., R. Huang, and W. Cheng. "An interval type-2 fuzzy-neural network with support-vector regression for noisy regression problems." *IEEE Trans on Fuzzy Systems* 18, no. 4 (2010): 686-699.

Juang, C., R. Huang, and Y. Lin. "A recurrent self-evolving interval type-2 fuzzy neural network for dynamic system processing." *IEEE Trans on Fuzzy Systems* 17, no. 5 (2009): 1092-1105.

Juang, C.F., and C.D. Hsieh. "A locally recurrent fuzzy neural network with support vector regression for dynamic-system modeling." *IEEE Trans on Fuzzy Systems* 18, no. 2 (2010): 261-273.

Juang, C.F., and C.T. Lin. "A recurrent self-organizing neural fuzzy inference network." *IEEE Trans on Neural Networks* 10, no. 4 (1999): 828-845.

Juang, C.F., and J.S. Chen. "A recurrent fuzzy-network-based inverse modeling method for a temperature system control." *IEEE Trans on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 37, no. 3 (2007): 410-417.

Kacprzyk, J., and S. Zadrozny. "Computing with words in intelligent database querying: standalone and Internet-based applications." *Information Sciences* 134, no. 1-4 (2001): 71-109.

Kangmao, W., C.K. Wang, and H. Chun. "Analytic hierarchy process with fuzzy scoring in evaluating multidisciplinary R&D projects in China." *IEEE Trans on Engineering Management* 52, no. 1 (2005): 119-129.

Karray, FO., and C.de Silva. *Soft Computing and Intelligent Systems Design.* Pearson Addison Wesley, 2004.

Kearfott, R.B., and V. Kreinovich. *Applications of interval computations.* Dordrecht: Kluwer, 1996.

Kennedy, J., and R.C. Eberhart. "Particle swarm optimization." *IEEE International Conference on Neural Networks.* NJ, 1995. 1942-1948.

Kennedy, J., and R.C. Eberhart. "Swarm Intelligence." *San Mateo, CA:Morgan Kaufmann*, 2001.

Khoshgoftaar, T.M., E.B. Allen, J.P. Hudepohl, and S.J. Aud. "Application of neural networks to software quality modeling of a very large telecommunications system." *IEEE Trans on Neural Networks* 8, no. 4 (1997): 902-909.

Kim, K., and S. Cho. "Evolutionary ensemble of diverse artificial neural networks." *Neurocomputing* 71, no. 7-9 (2008): 1604-1618.

Kirpatrick, S., C.D. Gelatt, and M.P. Vecchi. "Optimization by simulated." *Science* 220, no. 4598 (1983): 671-680.

Klement, E.P., R. Mesiar, and E. Pap. *Triangular Norms.* Dordrecht: Kluwer, 2002.

Korpela, J., and M. Tuominen. "Benchmarking logistics performance with an application of the analytic hierarchy process." *IEEE Trans on Engineering Management* 43, no. 3 (1996): 323-333.

Korzybski, A. *Science and Sanity.* Lakeville: Int. Non-Aristotelian Library Publishing Company, 1933.

Kreinovich, V., A. Lakeyev, J. Rohn, and P. Kahl. *Computational complexity and feasibility of data processing and interval computations.* Dordrecht: Kluwer, 1998.

Kuncheva, L.I., and J.J. Rodriguez. "Classifier ensembles with a random linear oracle." *IEEE Trans on Knowledge and Data Engineering* 19, no. 4 (2007): 500-508.

Kuo, R.J., P. Wu, and C.P. Wang. "An intelligent sales forecasting system through integration of artificial neural networks and fuzzy neural networks with fuzzy weight elimination." *Neural Networks* 15, no. 7 (2002): 909-925.

Leong, W., and G.G. Yen. "PSO-based multiobjective optimization with dynamic population size and adaptive local archives." *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics* 38, no. 5 (2008): 1270-1293.

Li, X. "A non-dominated sorting particle swarm optimizer for multiobjective optimization." *Proc. Genetic. Evol. Comput. Conf.* 2003. 37-48.

Li, X., and J. Shen. "LMI approach for stationary oscillation of interval neural networks with discrete and distributed time-varying delays under impulsive perturbations." *IEEE Trans on Neural Networks* 21, no. 10 (2010): 1555-1563.

Liang, X., and W. Pedrycz. "Logic-based fuzzy networks: A study in system modeling with triangular norms and uninorms." *Fuzzy Sets and Systems* 160, no. 24 (2009): 3475-3502.

Lin, F.J., P.K. Huang, C.C. Wang, and L.T. Teng. "An induction generator system using fuzzy modeling and recurrent fuzzy neural network." *IEEE Trans on Power Electronics* 22, no. 1 (2007): 260-271.

Lin, Y., and G.A. Cunningham. "A new approach to fuzzy-neural system modeling." *IEEE Trans on Fuzzy Systems* 3, no. 2 (1995): 190-198.

Ling, S.H., H.H.C. Iu, K.Y. Chan, H.K. Lam, B.C.W. Yeung, and F.H. Leung. "Hybrid particle swarm optimization with wavelet mutation and its industrial applications." *IEEE Trans on Systems, Man and Cybernetics, Part B:Cybernetics* 38, no. 3 (2008): 743-763.

Liu, P., and H. Li. "Efficient learning algorithms for three-layer regular feedforward fuzzy neural networks." *IEEE Trans on Neural Networks* 15, no. 3 (2004): 545-558.

Liu, Z., and H. Li. "A probabilistic fuzzy logic system for modeling and control." *IEEE Trans on Fuzzy Systems* 13, no. 6 (2005): 848-859.

Marmarelis, V.Z., and X. Zhao. "Volterra models and three-layer perceptrons." *Neural Networks* 8 (1997): 1421-1433.

Mata, F., L. Martinez, and E. Herrera-Viedma. "An adaptive consensus support model for group decision-making problems in a multigranular fuzzy linguistic context." *IEEE Trans on Fuzzy Systems* 17, no. 2 (2009): 279-290.

McCauley-Bell, P., and A.B. Badiru. "Fuzzy modeling and analytic hierarchy processing to quantify risk leveles associated with occupational injuries—part I: the development of fuzzy –linguistic risk levels." *IEEE Trans on Fuzzy Systems* 4, no. 2 (1996): 124-131.

McCulloch, W., and W. Pitts. "A logical calculus of the ideas immanent in nervous activity." *Bull Math Biophys* 5, no. 4 (1943): 115-133.

Melin, P., O. Mendoza, and O. Castillo. "Face recognition with an improved interval type-2 fuzzy logic sugeno integral and modular neural networks." *IEEE Trans on Systems, Man and Cybernetics, Part A: Systems and Humans* 41, no. 5 (2011): 1001-1012.

Mendes, R., J. Kennedy, and J. Neves. "The fully informed particle swarm: Simpler, maybe better." *IEEE Trans on Evol. Comput.* 8, no. 3 (2004): 204-210.

Mostaghim, S., and J. Teich. "The role of ε-dominance in multi-objective particle swarm optimization methods." *Proc. Congr. Evol. Comput.* Canberra, 2003. 1764-1771.

Musilek, P., E. Pelikan, T. Brabec, and M. Simunek. "Recurrent neural network based gating for natural gas load prediction systems." *Int. J. Conference on Neural Networks.* Vancouver, 2006. 3736-3741.

Musilek, P., R. Guanlao, and G. Barreiro. "Genetic programming of fuzzy aggregation operations." *Journal of Intelligent and Fuzzy Systems* 16, no. 2 (2005): 107-118.

Mustafa, M.A., and J.F. Al-Bahar. "Project risk assessment using the analytic hierarchy process." *IEEE Trans on Engineering Management* 38, no. 1 (1991): 46-52.

Nabavi-Kerizi, S.H., M. Abadi, and E. Kabir. "A PSO-based weighting method for linear combination of neural networks." *Computers and Electrical Engineering* 36, no. 5 (2010): 886-894.

Nobuhara, H., W. Pedrycz, S. Sessa, and K. Hirota. "A motion compression/reconstruction method based on max t-norm composite fuzzy relational equations." *Information Sciences* 176, no. 17 (2006): 2526-2552.

Nola, A. Di., G. Georgescu, and A. Iorgulescu. "Pseudo BL-algebras: Part I." *Mult.-Valued Log.* 8 (2002): 673-714.

Oh, S.K., W. Pedrycz, and S.B. Roh. "Hybrid fuzzy set-based polynomial neural networks and their development with the aid of genetic optimization and information granulation." *Applied Soft Computing* 9, no. 3 (2009): 1068-1089.

Panoutsos, G., and M. Mahfouf. "A neural-fuzzy modeling framework based on granular computing: concepts and applications." *Fuzzy Sets and Systems* 161, no. 21 (2010): 2808-2830.

Park, H., W. Pedrycz, and S. Oh. "Granular neural networks and their development through context based clustering and adjustable dimensionality of receptive fields." *IEEE Trans on Neural Networks* 20, no. 10 (2009): 1604-1616.

Pasi, G., and R.R. Yager. "Modeling the concept of majority opinion in group decision making." *Information Sciences* 176, no. 4 (2006): 390-414.

Pawlak, Z. "Rough sets." *Int. J. of Computer and Information Sciences* 11 (1982): 341-356.

—. *Rough Sets: Theoretical Aspects of Reasoning about Data.* Dordrecht: Kluwer Academic Publishers, 1991.

Pawlak, Z., and A. Skowron. "Rough sets and Boolean reasoning." *Information Sciences* 177, no. 1 (2007): 41-73.

Pawlak, Z., and A. Skowron. "Rough sets: Some extensions." *Information Sciences* 177, no. 1 (2007): 28-40.

Pawlak, Z., and A. Skowron. "Rudiments of rough sets." *Information Sciences* 177, no. 1 (2007): 3-27.

Pedrycz, W. *Computational Intelligence: An Introduction.* CRC Press LLC, 1998.

Pedrycz, W. "From fuzzy sets to shadowed sets: interpretation and computing." *International Journal of Intelligent Systems* 24, no. 1 (2009): 48-61.

—. *Granular Computing: An emerging paradigm.* Physica-Verlag, 2001.

—. *Knowledge-Based Clustering: From Data to Information Granules.* NJ: Wiley, 2005.

Pedrycz, W. "Logic-based fuzzy neurocomputing with unineurons." *IEEE Trans on Fuzzy Systems* 14, no. 6 (2006): 860-873.

Pedrycz, W. "Shadowed sets: representing and processing fuzzy sets." *IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics* 28 (1998): 103-109.

Pedrycz, W., and F. Gomide. *Fuzzy Systems Engineering: toward human-centric computing.* John Wiley and Sons, 2007.

Pedrycz, W., and G. Vukovich. "Granular neural networks." *Neurocomputing* 36, no. 1-4 (2001): 205-224.

Pedrycz, W., and K. Hirota. "A consensus-driven clustering." *Pattern Recognition Letters* 29, no. 9 (2008): 1333-1343.

Pedrycz, W., and P. Rai. "A multifacted perspective at data analysis: a study in collaborative intelligent agents." *IEEE Trans on Systems, Man, and Cybernetics, Part B:Cybernetics* 38, no. 4 (2008): 1062-1072.

Pedrycz, W., and P. Rai. "Collaborative clustering with the use of Fuzzy C-Means and its quantification." *Fuzzy Sets and Systems* 159, no. 18 (2008): 2399-2427.

Pedrycz, W., and R.A. Aliev. "Logic-oriented neural networks for fuzzy neurocomputing." *Neurocomputing* 73, no. 1-3 (2009): 10-23.

Pedrycz, W., B. Russo, and G. Succi. "Knowledge transfer in system modeling and its realization through an optimal allocation of information granularity." *Applied Soft Computing*, 2012.

Pedrycz, W., H.S. Park, and S.K. Oh. "A granular-oriented development of functional radial basis function neural networks." *Neurocomputing* 72, no. 1-3 (2008): 420-435.

Pelaez, J.I., and M.T. Lamata. "A new measure of consistency for positive reciprocal matrices." *Computers and Mathematics with Applications* 46, no. 12 (2003): 1839-1845.

Perez, I.J., F.J. Cabrerizo, and E. Herrera-Viedma. "A mobile decision support system for dynamic group decision making problems." *IEEE Trans on Systems, Man and Cybernetics, Part A:Systems and Humans* 40, no. 6 (2010): 1244-1256.

Pham, D.T., and M. Castellani. "Evolutionary learning of fuzzy models." *Applications of Artificial Intelligence* 19, no. 6 (2006): 583-592.

Piuri, V. "Analysis of fault tolerance in artificial neural networks." *Journal of Parallel and Distributed Computing* 61, no. 1 (2001): 18-48.

Prodanovic, P., and S.P. Simonovic. "Fuzzy compromise programming for group decision making." *IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans* 33, no. 3 (2003): 358-365.

Ratnaweera, A., S.K. Halgamuge, and H.C. Waston. "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients." *IEEE Trans on Evol. Comput.* 8, no. 3 (2004): 240-255.

Reformat, M. "A fuzzy-based multimodel system for reasoning about the number of software defects." *International Journal of Intelligent Systems* 20, no. 11 (2005): 1093-1115.

Reformat, M., and R.R. Yager. "Building ensemble classifiers using belief functions and OWA operators." *Soft Computing* 12, no. 6 (2008): 543-558.

Ridella, S., S. Rovetta, and R. Zunino. "Circular backpropagation networks for classification." *IEEE Trans on Neural Networks* 8, no. 1 (1997): 84-97.

Rocha, M., P. Cortez, and J. Neves. "Evolutionary design of neural networks for classification and regression." *Adaptive and Natural Computing algorithm*, 2005: 304-307.

Roubos, H., and M. Setnes. "Compact and transparent fuzzy models and classifiers through iterative complexity reduction." *IEEE Trans on Fuzzy Systems* 9, no. 4 (2001): 516-524.

Rumelhart, D.E., and J.L. McClelland. *Parallel Distributed.* Cambridge MA: MIT Press, 1986.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. "Learning representations by back-propagating errors." *Nature* 323 (1986): 533-536.

Saaty, T.L. "A new macroeconomic forecasting and policy evaluation method using the analytic hierarchy process." *Mathematical Modelling* 9 (1987): 219-231.

Saaty, T.L. "Axiomatic foundation of the analytic hierarchy process." *Management Science* 32, no. 7 (1986): 841-855.

Saaty, T.L. "How to handle dependence with the analytic hierarchy process." *Mathematical Modelling* 9 (1987): 369-376.

Saaty, T.L. "Introduction to a modeling of social decision process." *Mathematics and Computers in Simulation* 25 (1983): 105-107.

—. *The analytic hierarchy process, planning, priority setting, resource allocation.* New York: McGraw-Hill, 1980.

Saaty, T.L., and G. Hu. "Ranking by eigenvector versus other methods in the analytic hierarchy process." *Applied Mathematics Letters* 11, no. 4 (1998): 121-125.

Saaty, T.L., and M. Ozdemir. "Negative priorities in the analytic hierarchy process." *Mathematical Modelling* 37, no. 9-10 (2003): 1063-1075.

Schapire, R. "Strength of weak learnability." *Machine Learning* 5, no. 2 (1990): 197-227.

Shi, Y., and R. Eberhart. "A modified particle swarm optimizer." *Proc. IEEE Int. Conf. Evol. Comput.* 1998. 69-73.

—. "Empirical study of particle swarm optimization." *Proc.IEEE Congr.Evol. Comput.* 1999. 1945-1950.

—. "Parameter selection in particle swarm optimization." *Proc. 7th Int. Conf. Evol. Program.* 1998. 591-600.

Shumway, R.H. *Applied statistical time series analysis.* Englewood Cliffs, NJ:Prentice Hall, 1988.

Skowron, A. "The relationship between the rough set theory and evidence theory." *Bull Pol. Acad. Sci., Tech.* 37 (1989): 87-90.

Stadtherr, M.A., C.A. Schnepper, and J.F. Brennecke. "Robust phase stability analysis using interval methods." *International Symposium on Foundations of Computer aided Process Design.* 1994.

Stylios, C.D., and P.P. Groumpos. "Modeling complex systems using fuzzy cognitive maps." *IEEE Trans on Systems, Man, and Cybernetics, Part A: Systems and Humans* 34, no. 1 (2004): 155-162.

Sunaga, T. *Theory of interval algebra and its applications to numerical analysis.* Tokyo: Gaukutsu Bunken Fukeyu-kai, 1958.

Szczypinska, A., and E.W. Piotrowski. "Inconsistency of the judgment matrix in the AHP method and the decision maker’s knowledge." *Physica A: Statistical Mechanics and its Applications* 388 (2009): 907-915.

Szymanski, P.T., M. Lemmon, and C.J. Bett. "Hybrid interior points training of modular neural networks." *Neural Networks* 11, no. 2 (1998): 215-234.

Trelea, I.C. "The particle swarm optimization algorithm: convergence analysis and parameter selection." *Information Processing Letters* 85, no. 6 (2003): 317-325.

Tripathi, P.K., S. Bandyopadhyay, and S.K. Pal. "Multi-Objective Particle Swarm Optimization with time variant inertia and acceleration coefficients." *Information Sciences* 177, no. 22 (2007): 5033-5049.

Tung, S.L., and S.L. Tang. "A comparison of the Saaty’s AHP and modified AHP for right and left eigenvectors inconsistency." *European Journal of Operational Research* 106, no. 1 (1998): 123-128.

van Laarhoven, P.J.M., and W. Pedrycz. "A fuzzy extension of Saaty's priority theory." *Fuzzy Sets and Systems* 11, no. 1-3 (1983): 229-241.

Velez, M.A., O. Sanchez, S. Romero, and J.M. Andujar. "A new methododology to improve interpretability in neuro-fuzzy TSK models." *Applied Soft Computing* 10, no. 2 (2010): 578-591.

Wang, C., C. Cheng, and T. Lee. "Dynamical optimal training for interval type-2 fuzzy neural network (T2FNN)." *IEEE Trans on Systems, Man and Cybernetics, Part B: Cybernetics* 34, no. 3 (2004): 1462-1477.

Wang, L., and R. Langari. "Complex systems modeling via fuzzy logic." *Proceedings of the 33rd IEEE Conference on Decision and Control.* 1994. 4136-4141.

Wang, S., and J. Watada. "A hybrid modified PSO approach to VaR-based facility location problems with variable capacity in fuzzy random uncertainty." *Information Sciences*, 2010.

Wang, T.C., and Y.H. Chen. "Applying fuzzy linguistic preference relations to the improvement of consistency of fuzzy AHP." *Information Sciences* 178, no. 19 (2008): 3755-3765.

Warmus, M. "Approximations of inequalities in the calculus of approximations: Classification of approximate numbers." *Bulletin de l'Academie Polonaise des Sciences* 9, no. 4 (1961): 241-245.

Warmus, M. "Calculus of approximations." *Bulletin de l'Academie Polonaise des Sciences* 4, no. 5 (1956): 253-259.

Wedge, D., D. Ingram, D. McLean, C. Mingham, and Z. Bandar. "On global–local artificial neural networks for function approximation." *IEEE Trans on Neural Networks* 17, no. 4 (2006): 942-952.

Wiener, N. "On the nature of mathematical thinking." *Austrilian J. of Psychology and Philosophy* 1, no. 4 (1923): 268-272.

Williams, R.H. *Electrical Engineering Probability.* West Pub. (St. Paul), 1991.

Wu, D., and J.M. Mendel. "Aggregation using the linguistic weighted average and interval type-2 fuzzy sets." *IEEE Trans on Fuzzy Systems* 15, no. 6 (2007): 1145-1161.

Xu, Z., and R.R. Yager. "Power-geometric operators and their use in group decision making." *IEEE Trans on Fuzzy Systems* 18, no. 1 (2010): 94-105.

Yager, R.R. "Hierarchical aggregation functions generated from belief structures." *IEEE Trans on Fuzzy Systems* 8, no. 5 (2000): 481-490.

Yager, R.R. "On ordered weighted averaging aggregation operators in multicriteria decision making." *IEEE Trans on Systems, Man, Cybernetics* 18, no. 1 (1988): 183-190.

Yeh, C.Y., W.R. Jeng, and S.J. Lee. "Data-based system modeling using a type-2 fuzzy neural network with a hybrid learning algorithm." *IEEE Trans on Neural Networks* 22, no. 12 (2011): 2296-2309.

Zadeh, L.A. "Fuzzy sets and information granularity, in: M.M. Gupta, R.K. Ragade, R.R. Yager (Eds.)." *Advances in Fuzzy Set Theory and Applications*, 1979: 3-18.

Zadeh, L.A. "Is there a need for fuzzy logic?" *Information Sciences* 178, no. 13 (2008): 2751-2779.

Zadeh, L.A. "The role of fuzzy logic in the management of uncertainty in expert systems." *Fuzzy Sets and Systems* 11, no. 1-3 (1983): 199-227.

Zadeh, L.A. "Toward a generalized theory of uncertainty (GTU) — an outline." *Information Sciences* 172, no. 1-2 (2005): 1-40.

Zadeh, L.A. "Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic." *Fuzzy Sets and Systems* 90, no. 2 (1997): 111-117.

Zadrozny, S., and J. Kacprzyk. "Computing with words for text processing: An approach to the text categorization." *Information Sciences* 176, no. 4 (2006): 415-437.

Zhang, H., Z. Wang, and D. Liu. "Robust stability analysis for interval cohen-grossberg neural networks with unknown time-varying delays." *IEEE Trans on Neural Networks* 19, no. 11 (2008): 1942-1955.

Zhang, L.B., C.G. Zhou, X.H. Liu, Z.Q. Ma, M. Ma, and Y.C. Liang. "Solving multi objective problems using particle swarm optimization." *Proc. Congr. Evol. Comput.* Canberra, 2003. 2400-2405.

Zhang, X., and Q. Han. "Global asymptotic stability for a class of generalized neural networks with interval time-varying delays." *IEEE Trans on Neural Networks* 22, no. 8 (2011): 1180-1192.

Zhang, Y.Q., B. Jin, and Y. Tang. "Granular neural networks with evolutionary interval learning." *IEEE Trans on Fuzzy Systems* 16, no. 2 (2008): 309-319.

Zhang, Y.Q., M.D. Fraser, R.A. Gagliano, and A. Kandel. "Granular neural networks for numerical-linguistic data fusion and knowledge discovery." *IEEE Trans on Neural Networks* 11, no. 3 (2000): 658-667.

Zhao, L., F. Qian, Y. Yang, Y. Zeng, and H. Su. "Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization." *Applied Soft Computing* 10, no. 3 (2010): 938-944.

Zhou, Z.H., and Y. Yu. "Ensembling local learners through multimodal perturbation." *IEEE Trans on Systems, Man and Cybernetics, Part B: Cybernetics* 35, no. 4 (2005): 725-735.

Zitzler, E., K. Deb, and L. Thiele. "Comparison of multiobjective evolutionary algorithms: Empirical results." *Evolutionary Computation* 8, no. 2 (2000): 173-195.