Scientia Iranica D (2011) 18 (3), 617–626

Sharif University of Technology

Scientia Iranica

Transactions D: Computer Science & Engineering and Electrical Engineering

www.sciencedirect.com

Perception-based heuristic granular search: Exploiting uncertainty for analysis of certain functions

A. Rowhanimanesh, M.R. Akbarzadeh-T *

Cognitive Computing Lab., Center for Applied Research on Soft Computing and Intelligent Systems (CARSIS), Ferdowsi University of Mashhad, Mashhad, Iran

Received 12 September 2010; revised 26 October 2010; accepted 14 February 2011

KEYWORDS

Uncertainty;
Optimization;
Search;
Granular computing;
Function analysis;
Extension principle;
Certain functions;
Likeliness of inaccuracy.

Abstract Conventional approaches to optimization generally utilize a point-based search to scan domains of complex functions. These optimization algorithms, as a result, face a perpetual search that is never concluded with certainty, since the search space can never be completely scanned. In contrast, the proposed approach benefits from a granular view to scan the whole of the domain space. Such perspective can yield an efficient tool for analysis of complex functions, especially when proof is required. In contrast to conventional granular techniques that usually compute with certain granules, this scheme exploits uncertain granules, in addition to certain ones, to improve computational efficiency. To efficiently navigate the search space, Zadeh's extension principle, along with several heuristics, is introduced to estimate and reduce the likeliness of inaccuracy. Function analysis is then converted to a question–answering process. This method is general and can be applied to all types of functions whether linear or nonlinear, analytical or non-analytical and continuous or discrete. Several examples and a MATLAB toolbox are provided to illustrate the real-world applicability and computational efficiency of the approach.

1. Introduction

Analysis of functions plays a pivotal role in nearly all fields of engineering and other sciences, like optimization, control, modeling, analysis and design. Thus, increasing the efficiency of function analysis methods is an important research area, especially for nonlinear, high-dimensional and complex functions. Most conventional methods apply a point-wise view to scan the function's domain space point by point. Since a continuous domain space has an infinite number of points, conventional methods could never scan the whole of the domain space, especially for complex functions. In comparison with conventional techniques, the proposed approach uses a novel granular view to scan the domain space of a function, granule by granule. Using this approach, all points of the domain space of the function could be scanned without skipping even one point. Thus, this method is an efficient tool for analysis of nonlinear and complex functions when proof is required.

In the following, the earlier works by authors are briefly considered. Akbarzadeh-T et al. in 2007–2008 [1–3] focused on the application of fuzzy granulation to replace fitness function computation in evolutionary optimization. Authors, in 2009 [4–6], applied this approach to the extension principle in order to increase the performance of evolutionary optimization. In continuation, we further propose here the use of several heuristics for estimation and reduction of the likelihood of inaccuracy of the proposed EP. Also, the proposed approach is extended to all function analysis problems and further discussions and examples are presented. Moreover, an open source toolbox [7] is introduced for real-world applications of the proposed scheme.

Some earlier works that are directly or indirectly related to the topic of this paper are briefly reviewed. Pedrycz in 2010 [8] simultaneously, used granular computing and evolutionary optimization, and discussed their synergism for proposing an approach to designing cognitive maps. Tao et al. in 2008 [9] discussed the evolutionary characteristics of knowledge granulation and proposed the Evolutionary Algorithm of Knowledge Granulation (EAKG), which applies knowledge granulation to genetic programming. Batyrshin and Shereme-

* Corresponding author.
E-mail address: akbarzadeh@ieee.org (M.R. Akbarzadeh-T).

doi:10.1016/j.scient.2011.04.015

Production and hosting by Elsevier

tov [10] employed CWP and granular computing to propose a novel approach to perception-based data mining. Qin et al. [11] applied CWP to design a PNL-based question–answering machine. Gacek and Pedrycz in 2006 [12] applied granular computing for granular description of ECG signals. Reformat and Pedrycz in 2001 [13] proposed a genetic-based approach for evolutionary optimization of information granules. Ho and Lee in 2000 [14] presented a search algorithm, granular optimization, as an approach to functional optimization.

The main differences between the proposed scheme and other granular methods are as follows. Most of the former granular methodologies were designed for a special application or a typical class of functions, while this paper proposes a general approach to function analysis that could be applied to all types of function, including linear or nonlinear, analytical or non-analytical and continuous or discrete. This is achieved by applying Zadeh's generalized extension principle and using its approximate form to reduce its computational complexity. This approach is further enhanced by applying several heuristics that exploit the ambiguity in the likeliness of a granule to contain the solution. Also, it should be mentioned that the proposed scheme is based on fuzzy granulation and, therefore, crisp granulation (interval-based analysis), as used in this paper, is only a special case of it. Furthermore, this approach can be generalized for analysis of uncertain functions, when the known parts of the function are considered as certain and the uncertain parameters are considered as additional variables of the function.

This paper is organized as follows. The proposed granular approach is discussed in Section 2. Four generalized forms of the conventional Extension Principle (EP) are introduced here. Due to the computational complexity of the fourth generalized form, a new EP is proposed that is computationally efficient but only gives an approximate range of the granular value of the function over a given granule. Theorem 1 shows that this approximate range always contains the actual range. Then, several heuristics are proposed for estimation and reduction of the likelihood of inaccuracy of the proposed EP. The Perception-Based Problem Solver, a MATLAB toolbox, is introduced in Section 3, and the efficiency of the proposed approach is illustrated by several examples. Finally, conclusions are drawn in Section 4.

2. The proposed granular approach

In fuzzy computing [15–23], the Extension Principle (EP) plays a central role in granular analysis of certain functions. The basic form of EP is as follows:

$$\begin{cases} X \text{ is } A \\ y = f(X), \quad X = [x_1, x_2, \dots, x_n] \\ y \text{ is } B \end{cases}$$

$$\mu_A(X) : \text{known},$$

$$\mu_B(y) = ?$$

$$\overline{\mu_B(y) = \max_X(\mu_A(X))}$$

$$\text{s.t.} \quad y = f(X).$$

Suppose granule A (a fuzzy set) is given in the domain space and EP wishes to calculate the corresponding granular value of y. The method that has been introduced in [15] is usually used for solving EP. This method is based on discretization of y over its range, and then solving several n-dimensional nonlinear programming problems corresponding to discretized values of y. This means that solving EP leads to solving several n-dimensional nonlinear programming problems. It is clear that the computational efficiency of this approach is not satisfying when n is large and f is complex.

In the following sections, four general forms of EP are first introduced. It is shown that the above-mentioned complexity is only related to the fourth form of EP. Then, a new form of EP is proposed to be used instead of the fourth form. The proposed EP is computationally more efficient, but it only provides an approximate range that *contains* the actual granular value of y. Several heuristics are then suggested to improve the accuracy of the proposed EP.

2.1. General forms of extension principle

Figure 1 illustrates the four general forms of the extension principle that directly result from Zadeh's standard extension principle. In this figure, $\mu(.)$ is the possibility distribution function (membership function) of the constraining relation; x and X are a scalar variable and a vector of n variables, respectively; f and g are scalar functions, * stands for basic arithmetic operators including $+, -, \times, \div$; and $\hat{}$ is a t-norm operator. The aim is to calculate $\mu_{B_i}(y)$, i.e. the possibility distribution function of the output where $i = 1 : 4$ is the form index. Regarding Forms 1 to 4, calculating $\mu_{B_i}(y)$ leads to solving a constrained *maximization* problem [15]. As discussed further below, solving this maximization problem is trivial for Forms 1 to 3, but it is complex and computationally expensive for Form 4, when the number of variables is large.

Form 1 represents the standard extension principle if the maximization is performed over a scalar domain and for this reason, $\mu_{B1}(y)$ or at least an accurate approximation of it can be achieved, either explicitly or numerically. Form 2 shows the standard extension principle which is extended to scalar functions. Regarding Figure 1(b), calculating $\mu_{B2}(y)$ is similar to $\mu_{B1}(y)$ of Form 1. Form 3 deals with the arithmetic combination of two scalar multi-variable functions, $f(X_1)$ and $g(X_2)$, when their variables, X_1 and X_2, are two independent vectors. Regarding Figure 1(c), calculating $\mu_{B3}(y)$ leads to solving the following constrained one-dimensional maximization problem, whose details are shown in Figure 1(c):

$$\mu_{B3}(y) = \max_{f(X_1), g(X_2)} (\mu_{A_{31}}(f(X_1)) \wedge \mu_{A_{32}}(g(X_2))), \tag{1}$$

$$\text{s.t.} \quad y = f^* g(X_1, X_2). \tag{2}$$

The constraint in Eq. (2) can be simply inserted in Eq. (1), i.e. the objective function, as shown in Figure 2. In other words, the above constrained maximization problem can be converted to an unconstrained maximization problem which is more conveniently solved. Generally, calculating $\mu_{Bi}(y)$ is simple for Forms 1 to 3, since maximization is performed over a one-dimensional space, while Form 4 is different.

Form 4 deals with the arithmetic combination of two scalar multi-variable functions, $f(X)$ and $g(X)$, whose variables, in contrast to Form 3, are allowed to be mutual. Regarding Figure 1(d), calculating $\mu_{B4}(y)$ leads to solving an n-dimensional nonlinear constrained maximization problem, which may be complex and computationally expensive when the number of variables is large. One way is by quantizing y and then applying an optimization scheme to solve the mentioned maximization problem for each quantized value of y. Zadeh, in [15], recommended the use of neuro and evolution-based computing to achieve an accurate approximation of $\mu_{B4}(y)$. It should be

(a) Form 1:

$$\begin{cases} x \text{ is } A_1 \\ f(x) \text{ is } B_1 \end{cases}$$

$\mu_{A_1}(x)$: known,

$\mu_{B_1}(f(x)) = ?$

$$\mu_{B_1}(y) = \max_{x}(\mu_{A_1}(x))$$

s.t. $y = f(x)$

(b) Form 2:

$$\begin{cases} f(X) \text{ is } A_2 \\ g(f(X)) \text{ is } B_2 \end{cases}$$

$\mu_{A_2}(f(X))$: known,

$\mu_{B_2}(g(f(X))) = ?$

$$\mu_{B_2}(y) = \max_{f(X)}(\mu_{A_2}(f(X)))$$

s.t. $y = g(f(X))$

(c) Form 3:

$$\begin{cases} f(X_1) \text{ is } A_{31} \\ g(X_2) \text{ is } A_{32} \\ f^* g(X_1, X_2) \text{ is } B_3 \end{cases}$$

$* : +, -, \times, \div$

$\mu_{A_{31}}(f(X_1)), \mu_{A_{32}}(g(X_2))$: known

$\mu_{B_3}(f^* g(X_1, X_2)) = ?$

$$\mu_{B_3}(y) = \max_{f(X_1), g(X_2)}(\mu_{A_{31}}(f(X_1)) \wedge \mu_{A_{32}}(g(X_2)))$$

s.t. $y = f^* g(X_1, X_2)$

(d) Form 4:

$$\begin{cases} f(X) \text{ is } A_{41} \\ g(X) \text{ is } A_{42} \\ f^* g(X) \text{ is } B_4 \end{cases}$$

$* : +, -, \times, \div$

$\mu_{A_{41}}(f(X)), \mu_{A_{42}}(g(X))$: known

$\mu_{B_4}(f^* g(X)) = ?$

$$\mu_{B_4}(y) = \max_{X}(\mu_{A_{41}}(f(X)) \wedge \mu_{A_{42}}(g(X)))$$

s.t. $y = f^* g(X)$

Figure 1: Four generalized forms of conventional EP: Forms 1 and 2 correspond to 1-variable basic functions, and Forms 3 and 4 correspond to multi-variable functions for basic arithmetic operators.

$$\mu_{B_3}(y) = \max_{g(X_2)}(\mu_{A_{31}}(y \otimes g(X_2)) \wedge \mu_{A_{32}}(g(X_2)))$$

$$\begin{aligned} &* : + \Rightarrow \otimes : - \\ &* : - \Rightarrow \otimes : + \\ &* : \times \Rightarrow \otimes : \div \\ &* : \div \Rightarrow \otimes : \times \end{aligned}$$

Figure 2: Simplified form of Form 3.

mentioned that such an approach, as well as other similar approaches, can be computationally expensive when the number of variables is large.

In many cases, however, a precise value of $\mu_{B4}(y)$ may not be necessary, and $\mu_C(y)$ can be used instead of $\mu_{B4}(y)$, where $\mu_{B4}(y) \subset \mu_C(y)$. In the next subsection, we propose an approach for calculating such a $\mu_C(y)$, which is computationally simpler than Form 4 and can be used instead. The proposed approach becomes more efficient as this approximation, $\mu_{B4}(y) \approx \mu_C(y)$, becomes more accurate.

2.2. Atomic and composite elements of a function

Suppose $f(x_1, x_2, \ldots, x_n)$ is an n-variable scalar function that can be computationally described. Here, f is allowed to be continuous or discrete, analytical or non-analytical and linear or nonlinear. For example, f may be represented as an ordinary mathematical formula including a combination of polynomial and sinusoidal functions, or may be described as the source of a computer program that includes algorithmic and user-defined sub-functions. If f has not been described computationally, fuzzy or neuro systems or other similar methods could be applied to computationally model it. Hence, in the following, we assume that f is computationally described without loss of generality.

A general computational statement, which describes an n-variable scalar function, f, is composed of two basic types of elements including *atomic* and *composite elements*. Atomic elements are generally simple scalar functions like e^x, $\sin(x)$ and x^r that operate on a one-dimensional domain. Composite elements are composed of two or more atomic elements that are joined by basic arithmetic operators, such as $+, -, \times, \div$ or other similar operators. f can be represented as a combination of its atomic and composite elements. After this decomposition, generalized forms of EP can be directly applied to each of these basic elements, and EP works on these simple terms instead of f, which is more complex. Example 1, as below, illustrates this process.

Example 1. Consider the Peaks function over the domain space $-10 \le x_1, x_2 \le 10$,

$$\begin{aligned} y = f(x_1, x_2) &= 3(1 - x_1)^2 e^{-(x_1)^2 - (x_2+1)^2} \\ &+ \left(-2x_1 + 10x_1^3 + 10x_2^5\right) e^{-(x_1)^2 - (x_2)^2} \\ &- \frac{1}{3} e^{-(x_1+1)^2 - (x_2)^2}, \end{aligned}$$

where $f(x_1, x_2)$ is composed of seven atomic elements as below:

Term 1: $3(1 - x_1)^2 e^{-(x_1)^2}$,

Term 2: $e^{-(x_2+1)^2}$,

Term 3: $(-2x_1 + 10x_1^3) e^{-(x_1)^2}$,

Term 4: $e^{-(x_2)^2}$,

Term 5: $(10x_2^5) e^{-(x_2)^2}$,

Term 6: $e^{-(x_1)^2}$,

Term 7: $-\frac{1}{3} e^{-(x_1+1)^2}$.

These atomic terms are combined by summation and multiplication operators as follows:

$$\begin{aligned} y = {} & \text{Term 1}^*\text{Term 2} + \text{Term 3}^*\text{Term 4} \\ & + \text{Term 5}^*\text{Term 6} + \text{Term 7}^*\text{Term 2}. \end{aligned}$$

a
$$\begin{cases} f(X) \text{ is } A_{41} \\ g(X) \text{ is } A_{42} \\ f^{*}g(X) \text{ is } C \end{cases}$$

$* : +, -, \times, \div$

$\mu_{A_{41}}(f(X)), \mu_{A_{42}}(g(X))$: known

$\mu_C(f^{*}g(X)) = ?$

$$\mu_C(y) = \max_{f(X), g(X)} (\mu_{A_{41}}(f(X)) \wedge \mu_{A_{42}}(g(X)))$$

s.t. $y = f^{*}g(X)$

b

$$\mu_C(y) = \max_{g(X)} (\mu_{A_{41}}(y \otimes g(X)) \wedge \mu_{A_{42}}(g(X)))$$

$* : + \Rightarrow \otimes : -$
$* : - \Rightarrow \otimes : +$
$* : \times \Rightarrow \otimes : \div$
$* : \div \Rightarrow \otimes : \times$

Figure 3: (a) The proposed EP instead of Form 4. (b) Simplified form of (a).

All the above atomic elements are single-variable functions, and the composite elements are created by well-known arithmetic operators. It is important to note that a considerable amount of information can be easily achieved concerning these single-variable atomic elements by plotting them. The behavior of summation and multiplication operators is clear and can be plotted as well. Using this information, the first three generalized forms of EP, as well as the proposed EP, can be easily solved symbolically. For more details, the related MATLAB codes (M-files) can be downloaded from [7].

2.3. The proposed extension principle

As Figure 1 illustrates, the main difference between Forms 3 and 4 lies in the relation between the variables of functions f and g. In Form 3, these variables are independent and this leads to a one-dimensional maximization, while in Form 4, they are dependent and, for this reason, an n-dimensional maximization problem must be solved. Figure 3 illustrates the proposed form of the extension principle that can be used instead of Form 4. Regarding the concept of $\mu_C(y)$, f and g in Form 4 can be assumed to be independent, then the new form can be written as shown in Figure 3(a). Figure 3(a) indicates a new form of the extension principle, all parts of which are same as in Form 4, except for the maximization part which is similar to Form 3. Thus, calculating $\mu_C(y)$ of the new form is similar to calculating $\mu_{B3}(y)$ of Form 3, and leads to solving a one-dimensional maximization problem, which is computationally trivial. Similar to Figure 2, Figure 3(b) shows the simplified form of Figure 3(a). In the following theorem, we prove that $\mu_{B4}(y) \subset \mu_C(y)$. Therefore, the proposed form of Figure 3 can be efficiently used in place of Form 4. In the following sections, this new form along with Forms 1 to 3 is applied to reasoning.

Theorem 1. *The fuzzy set B_4 of Form 4 (in Figure 1) is a subset of the fuzzy set C of the new form (in Figure 3), i.e. $B_4 \subset C$.*

Proof. Let us define notion S_F as the support of fuzzy set F. Consider y_1 and y_2 to be arbitrary members of $S_{A_{41}}$ and $S_{A_{42}}$, respectively (i.e. $y_1 \in S_{A_{41}}, y_2 \in S_{A_{42}}$). It is then clear $f(X) \in S_{A_{41}}, g(X) \in S_{A_{42}}$. Let U_X be the local universe of discourse of the n-dimensional vector variable ($X \in U_X$). Regarding the above definitions, it is obvious that:

$$\{(f(X), g(X)) | X \in U_X\} \subset \{(y_1, y_2) | y_1 \in S_{A_{41}}, y_2 \in S_{A_{42}}\}. \quad (3)$$

According to Relation 3, the following statement proves that support of B_4 is a subset of the support of C:

$$\{v | \forall X \in U_X : v = f(X)^{*}g(X)\}$$
$$\subset \{w | w = y_1^{*}y_2 : y_1 \in S_{A_{41}}, y_2 \in S_{A_{42}}\} : S_{B4} \subset S_C. \quad (4)$$

Using Relation 3, and given any fixed y_0, we have:

$$\{(f(X), g(X)) | y_0 = f(X)^{*}g(X), X \in U_X\}$$
$$\subset \{(y_1, y_2) | y_0 = y_1^{*} y_2, y_1 \in S_{A_{41}}, y_2 \in S_{A_{42}}\}. \quad (5)$$

From Relations 3 and 5, we can conclude that:

$$\mu_{B_4}(y_0) = \max_X (\mu_{A_{41}}(f(X)) \wedge \mu_{A_{42}}(g(X)))$$
$$\leq \max_{y_1, y_2} (\mu_{A_{41}}(y_1) \wedge \mu_{A_{42}}(y_2)) = \mu_C(y_0),$$
$$\text{s.t.} \quad y_0 = f(X)^{*}g(X) = y_1^{*}y_2. \quad (6)$$

From Relations 4 and 6, we conclude that $\forall y : \mu_{B4}(y) \leq \mu_C(y)$ and, consequently, $B_4 \subset C$. □

Theorem 1 proves that fuzzy set B_4 is a subset of fuzzy set C. Fuzzy set B_4 is the actual granular value of y (actual range), while fuzzy set C is an approximate range that contains B_4. It is not trivial to calculate how much C is larger than B_4, since it depends on the behaviors of f and g over the given granule in the n-dimensional domain space. This means that although the new EP is computationally efficient, it entails a degree of uncertainty.

The proposed approach, based on several heuristics, considers the *likeliness* of the resulting inaccuracy versus the computational complexity of the exact method, i.e. conventional EP. Furthermore, we consider here crisp granulation (interval-based granulation), since it is computationally simpler than fuzzy granulation. There exists a notable number of function analysis problems that can be solved by crisp granulation and do not require the extra information provided by fuzzy granulation.

2.4. Estimation and reduction of the likeliness of inaccuracy

In this section, five different heuristics are proposed for reducing and estimating the likelihood of inaccuracy of the proposed EP when crisp granulation is used. The large value of this likeliness usually decreases the efficiency of the proposed scheme. The proposed approaches are as follows: increasing the resolution, fuzzy evaluation, well-defined terms, finding an approximate range using optimization techniques and increasing available knowledge.

2.4.1. Increasing the resolution

The likelihood of inaccuracy is usually reduced by increasing the resolution, which means using smaller instead of larger granules. This is a good approach for solving small-scale problems. Since decreasing the size of granules leads to increasing the number of granules and consequently the number of function evaluations, this technique is not computationally efficient for large-scale problems if it is applied separately. The other methods that are proposed below solve this problem.

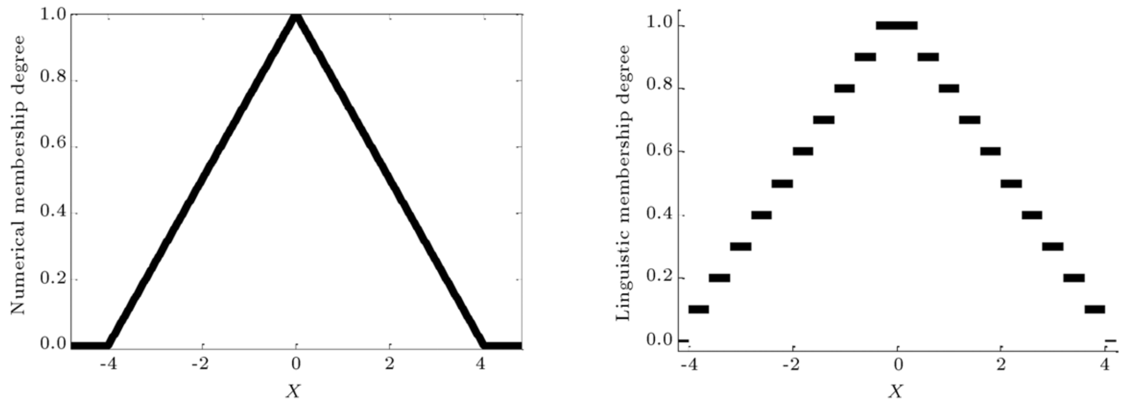

Figure 4: Conventional MF vs. interval-based MF for precisiating "x is about zero".

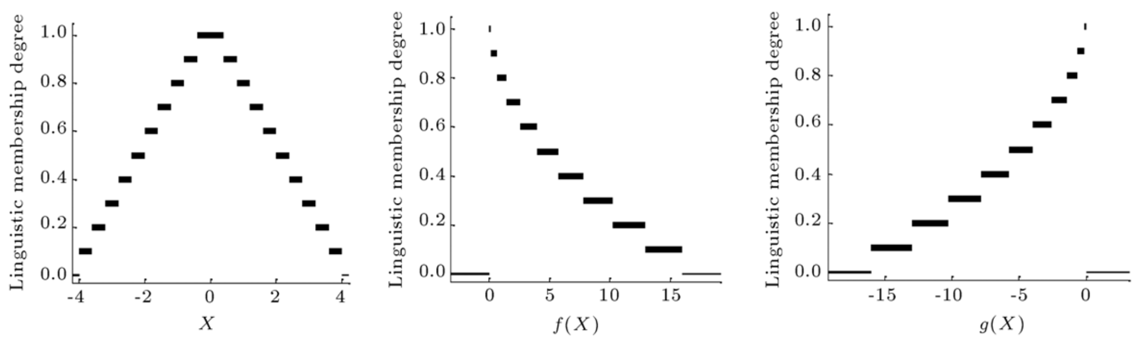

Figure 5: Fuzzy evaluation of the function of Example 2.

2.4.2. Fuzzy evaluation

In this approach, one or more fuzzy granules are defined over the given crisp granule in the domain space, and the proposed EP is separately solved over both crisp and defined fuzzy granules. These additional fuzzy evaluations provide useful knowledge about the behavior of the function over the given crisp granule. Although this knowledge can be applied for various purposes, here, it is used for estimating the likeliness of inaccuracy of the proposed EP over a given granule in the domain space. If the estimated value for the likeliness of inaccuracy is large, then the previous approach (increasing the resolution) can be applied to reduce inaccuracy. This approach is usually computationally efficient for most problems, even large-scale ones.

Due to computational simplicity, Interval-based Membership Functions (IMF) is used instead of conventional MF. Figure 4 introduces interval-based MF as an approximation of conventional MF. Conventional MF uses a precise conventional function for describing membership degree distribution, while such accuracy is not usually required, and it just increases computational complexity. In contrast, IMF uses an interval-based function for describing membership degree distribution, which is computationally efficient. Also, the accuracy of interval-based MF is usually acceptable.

The following example illustrates how fuzzy evaluation can be used for estimating the likeliness of inaccuracy of the proposed EP.

Example 2. What is the granular value of $f(x)+g(x)$ over $-4 \le x \le 4$ where $f(x) = x^2$ and $(x) = -x^2$?

Solution. Solving the proposed EP over $-4 \le x \le 4$ gives $[-16\ 16]$ as an approximate range of $f(x) + g(x)$, while the actual range is 0. For this example, the value of inaccuracy is very large. Figure 5 shows a fuzzy granule that is defined over $-4 \le x \le 4$ by a triangular interval-based MF, as well as the granular values of $f(x)$ and $g(x)$ over this fuzzy granule.

Regarding this figure, the membership degrees of both of $f(x)$ and $g(x)$ are about one when their values are about zero and also the membership degree of x is about one at zero. Thus, $f(x)$ and $g(x)$ are about zero, when x is about zero. If x moves away from zero, $f(x)$ increases and $g(x)$ decreases. Therefore, it is *highly unlikely* that the maximum value of $f(x)$ could be added to the maximum value of $g(x)$. The same result could be obtained for the minimum values. Consequently, it is *highly unlikely* that $[-16\ 16]$ is approximately equal to the actual range.

To systematize the above idea, sub-granules with the same membership degree are added. This idea comes from the behavior of EP, when an output point with a membership degree of q *certainly* relates to an input point with a membership degree of q; moreover, it *may* relate to other input points with membership degrees equal or less than q. Usually, the input sub-granule with membership degree of q is only known and it is not easy to determine if other sub-granules exist or not. Thus, decision making based on the above idea is *uncertain* rather than certain. In effect, the above idea helps machines to *guess* the value of the likeliness of inaccuracy of the proposed EP. By applying the above-mentioned idea to this example, Figure 6 is obtained. Comparing the range of Figure 6 (about $[-3\ 3]$) with $[-16\ 16]$ gives the following guess: "the likeliness of inaccuracy of the proposed EP over the given crisp granule is very high". Regarding the actual range, the precision of this guess is approved.

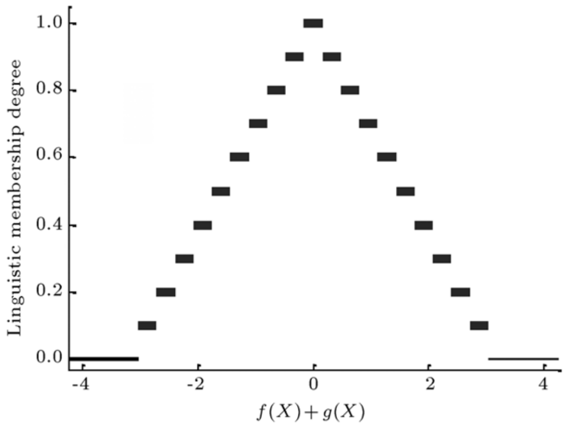

Figure 6: In Example 2, the sub-granules with the same membership degree are added.

2.4.3. Well-defined terms

Consider two terms (functions) that are combined by a combining operator. Without loss of generality, suppose that the combining operator is a summation. Assume that we aim to compute the range of the summation of two given terms over a given granule in the domain space. For the combining operator of summation, if both of these terms get their global maximums at the same point of the given granule, and also get their global minimums at another same point, then the range of summation of these terms over this granule is equal to the summation of the ranges of these terms over this granule. These terms are called well-defined for the summation operator over the given granule. In other words, if the range of each term could be calculated accurately, and if these terms are well-defined for the summation operator over the given granule, then the range of the summation of them over this granule could be computed accurately. It should be noted that these terms may not be *well-defined* over another granule in the domain space or for another combining operator.

Special but important classes of well-defined terms are the classes of increasing and decreasing functions, since an increasing function always gets its global minimum (maximum) at the smallest (largest) point of the given granule, and vice-versa for decreasing functions. The given two terms may be monotone over only a few granules in the domain space. The following example shows the details of this approach.

Example 3. Consider $g(x_1, x_2)$ as follows:

$$g(x_1, x_2) = \sin(0.25(x_1 + x_2))^4 + \left(\frac{2}{1 + e^{-2(x_1+x_2)}} - 1\right)^2,$$
$$\text{s.t.} \quad -3 \le x_1,\ x_2 \le 3.$$

$g(x_1, x_2)$ is the summation of $g_1(x_1, x_2) = \sin(0.25(x_1 + x_2))^4$ and $g_2(x_1, x_2) = (\frac{2}{1+e^{-2(x_1+x_2)}} - 1)^2$, which are not atomic, and each of them includes a few basic elements. By considering the defining formulas of g_1 and g_2, it can be observed that both are monotone around line $x_1 + x_2 = 0$. This claim can be proven from their partial derivatives. If the given granule does not include line $x_1 + x_2 = 0$, terms g_1 and g_2 are simultaneously increasing or decreasing. Thus, these terms are well-defined, and the range of g can be computed accurately over these granules. The concept of well-defined terms can be generalized to other combining operators.

2.4.4. Finding an approximate range using optimization techniques

Suppose the range of a function must be calculated over a given crisp granule. The proposed EP gives an approximate range that contains the actual range. Suppose that an optimization method like Genetic Algorithms is applied to find the minimum (m) and maximum (M) of the function over this granule. The obtained minimum and maximum values are usually near-optimal and thus the range $[m, M]$ is a subset of the actual range of the function over this granule. Thus, the actual range is limited between $[m, M]$ (as an approximation that is smaller than the actual range) and the approximate range obtained by EP (as an approximation that is larger than the actual range). This additional information could be used in different ways like estimating the likeliness of inaccuracy of the proposed EP.

2.4.5. Increasing available knowledge

Increasing available knowledge about the behavior of a function over the domain space could significantly help previous approaches in estimating and reducing the likeliness of inaccuracy. Indeed, fuzzy evaluation can be considered as a special case of increasing available knowledge. There exist different ways for exploiting this knowledge. For example, this knowledge could be employed for identifying semi-well-defined terms whose behavior is approximately similar to well-defined terms. Also, this knowledge could be applied to design an efficient termination criterion for the optimization technique used in the fourth approach. Moreover, it can be used for determining the size of smaller granules in each part of the space when resolution is increased.

Figure 7 shows the algorithmic flowchart of the proposed approach. In the next section, a MATLAB toolbox is introduced to implement this algorithm and few examples are solved to demonstrate the efficiency of the proposed scheme.

3. "Perception-based problem solver": an open-source toolbox for global search [7]

"Perception-based Problem Solver" is an open source toolbox in MATLAB for implementation of the proposed approach, and is available via [7]. This toolbox could be used as a question–answering machine for analysis of functions. Using the proposed granular approach, this toolbox scans the whole domain space of the given function without skipping even one point. Thus, it could be used as an efficient tool for analysis of nonlinear and complex systems, when proof is required. This toolbox uses the first approach of reducing inaccuracy, i.e. increasing the resolution. To increase its efficiency for solving high-dimensional problems, the other proposed heuristic approaches in the above section must be incorporated. The algorithm of this toolbox is described as follows.

First, user asks a question like the queries of Example 4. The original domain space is partitioned into a number of granules. This level is called *resolution level* 1. Then, the function is evaluated over each granule using the proposed approach. It should be noted that the proposed approach just gives an approximate range that contains the actual range. Depending on the given query, a number of granules are rejected certainly (*poor granules*) and the rest of them are kept. From the remained granules, a number of them certainly satisfy the query (*good granules*), which means that entire points of a good granule satisfy the query. The rest of the granules are likely granules, which are likely to satisfy the query, but it is not certain.

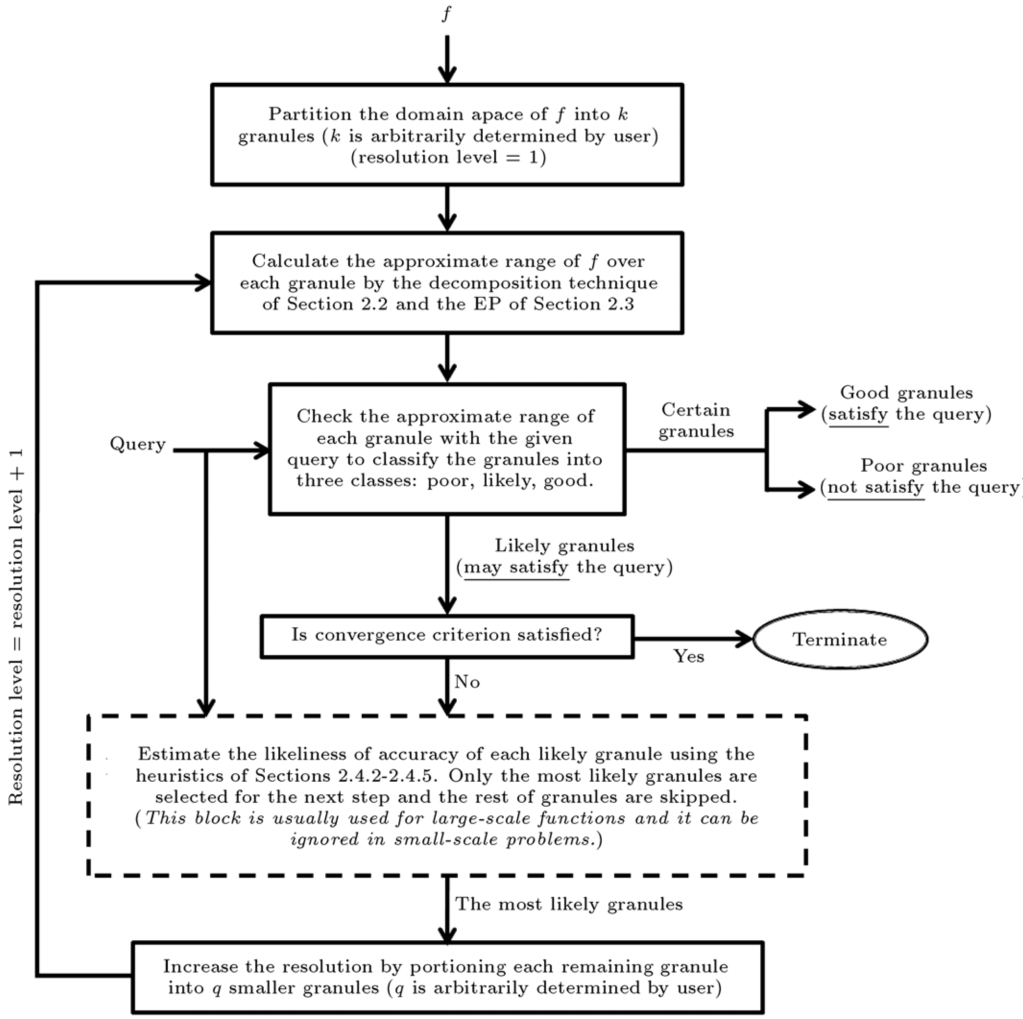

Figure 7: Flowchart of the proposed approach.

To certainly determine the status of likely granules, they are considered as the new domain space and are partitioned into smaller granules, then the above procedure is repeated for them. This level is called *resolution level* 2. The above procedure can be repeated several times. It is clear that the resolution is increased through resolution levels. As a convergence criterion, the search can be stopped at the current resolution level, if the size of all likely granules approaches the size of all good granules of the current resolution level.

Example 4. In this example, the proposed toolbox is used for analysis of Peaks function of Example 1. Several queries are asked by the user and answered by toolbox. In this example, at each resolution level, the domain of each variable over the given likely granule is partitioned into two equal parts. So each likely granule is partitioned into 4 equal parts. For all queries, the space is initially partitioned into 100 identical granules. The convergence factor is the ratio of the size of likely granules to the size of good granules. If the convergence factor approaches one, then the search is stopped.

Figure 8 shows the contour plot of the Peaks function. The answers that are provided by toolbox to the following different queries are shown in Figures 9–11. Comparing these answers

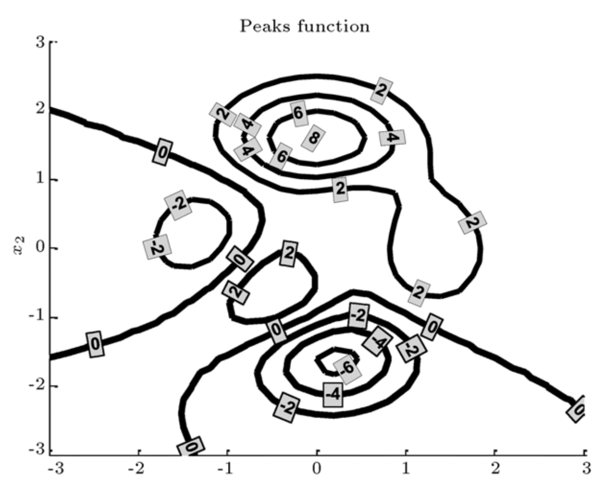

Figure 8: Visualization of Peaks function.

with the contour plot of the Peaks function (Figure 8) demonstrates the efficiency of the proposed approach. Interested readers can download this toolbox from [7] and run these examples on their own computers. Also, more solved examples are available in [7].

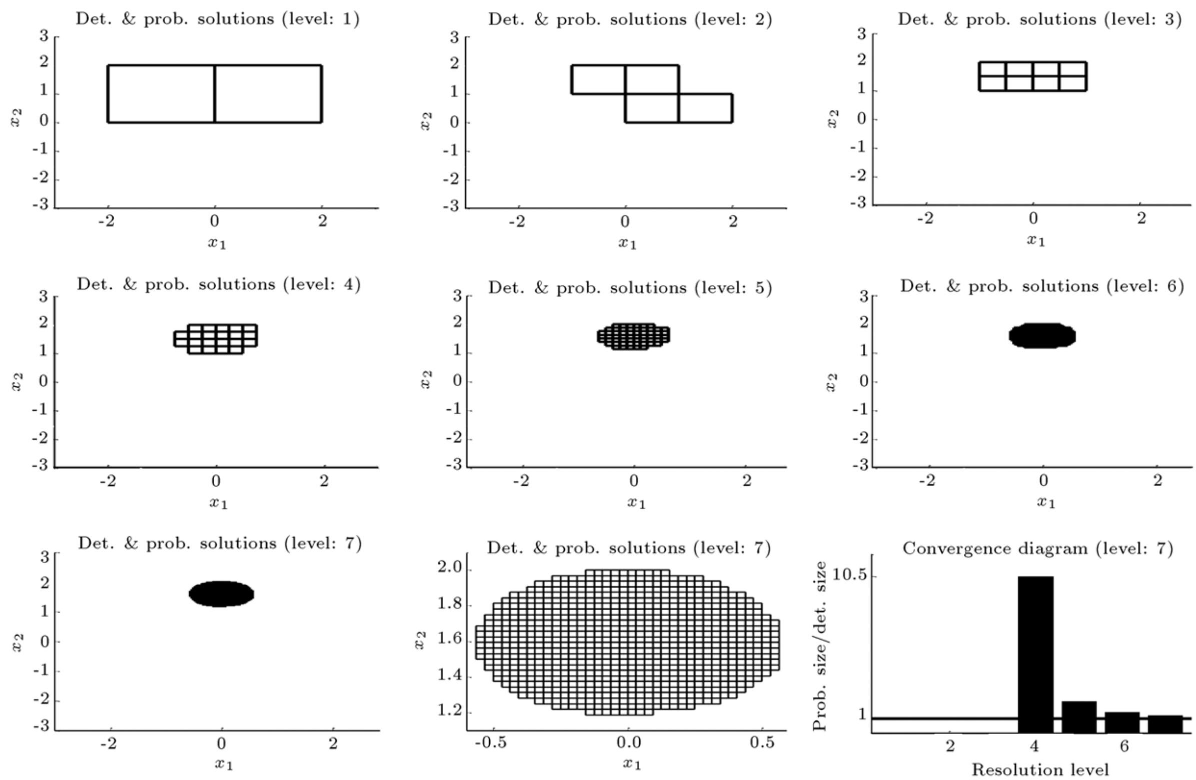

Figure 9: Query: In which parts of the domain space, $f(x_1, x_2)$ is larger than 6? The good granules (black) and likely granules (gray), as well as the convergence diagram, are shown at each resolution level. The first good solutions are found in resolution level 4.

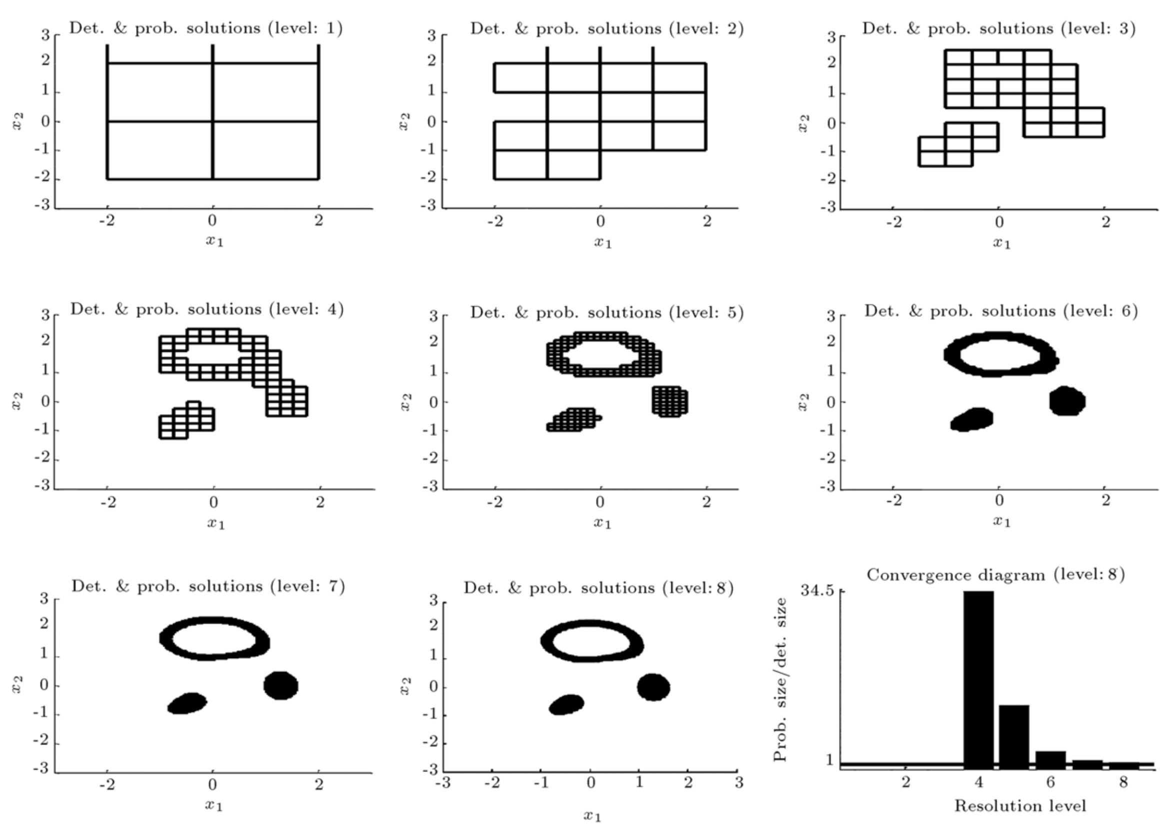

Figure 10: Query: In which parts of the domain space, $f(x_1, x_2)$ is between 3 and 4? The good granules (black) and likely granules (gray), as well as the convergence diagram, are shown at each resolution level. The first good solutions are found in resolution level 4.

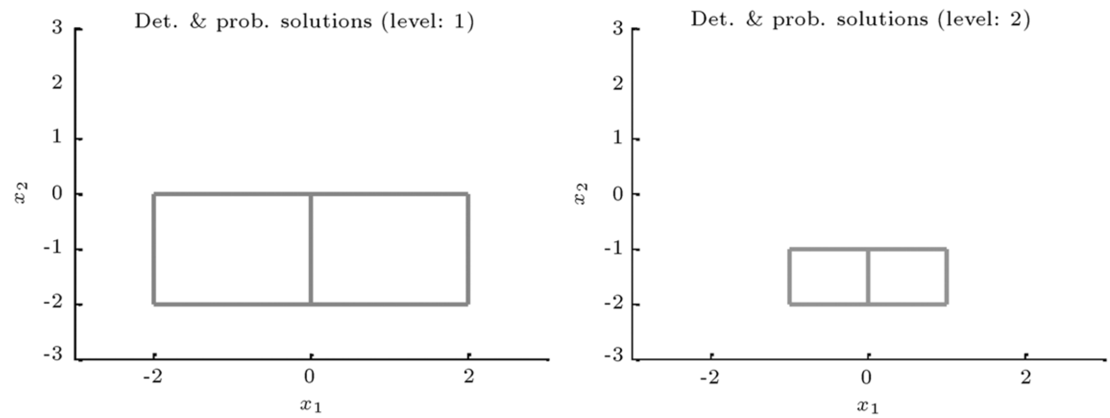

Figure 11: Query: In which parts of the domain space, $f(x_1, x_2)$ is smaller than -8? The likely granules (gray) are shown at each resolution level. No solution exists after resolution level 2.

4. Conclusion

This paper proposes a novel granular approach for analysis of certain functions based on exploiting uncertainty. This approach is general and can be applied to all types of function, whether linear or nonlinear, analytical or non-analytical, and continuous or discrete. Using this method, all points of the domain space of the function can be scanned without skipping even one point. This could be used as an efficient tool for analysis of complex functions, especially when proof is required. The proposed scheme effectively exploits uncertain granules to increase its computational efficiency. Several heuristics are suggested for estimation and reduction of the likeliness of inaccuracy. A MATLAB toolbox [7] is introduced for real-world applications of the proposed approach, and different examples are solved to show its efficiency. The proposed theorem applies to fuzzy granules and thus this approach can be generalized to fuzzy granulation. Also, the proposed scheme is flexible enough to be generalized for analysis of uncertain functions when known parts of the function are considered certain, and the uncertain parameters are considered additional variables of the function.

References

[1] Akbarzadeh-T, M.-R., et al. "Adaptive fuzzy fitness granulation for evolutionary optimization", *International Journal of Approximate Reasoning*, 49, pp. 523–538 (2008).
[2] Davarynejad, M., et al. "Auto-tuning fuzzy granulation for evolutionary optimization". In *2008 Congress on Evolutionary Computation*, CEC'2008, pp. 3573–3580, IEEE Service Center, Hong Kong (2008).
[3] Davarynejad, M., et al. "A novel general framework for evolutionary optimization: Adaptive fuzzy fitness granulation", *Proceedings of the 2007 IEEE International Conference on Evolutionary Computing*, CEC'2007, Singapore, pp. 951–956 (2007).
[4] Rowhanimanesh, A. and Akbarzadeh-T, M.-R. "Human-like rational exploration in evolutionary optimization using computing with words", *Proceedings of ICEE 2009, 17th Iranian Conference on Electrical Engineering*, Tehran, Iran (2009).
[5] Rowhanimanesh, A. and Akbarzadeh-T, M.-R. "Reducing the search space in evolutionary optimization using computing with words and perceptions", *Proceedings of IFS2009, the 3rd Joint Congress on Intelligent and Fuzzy Systems*, Yazd, Iran (2009).
[6] Rowhanimanesh, A. and Akbarzadeh-T, M.-R. "Perception-based evolutionary optimization: Outline of a novel approach to optimization and problem solving", *Proceedings of 2010 IEEE International Conference on Systems, Man and Cybernetics, SMC2010*, Istanbul, Turkey, pp. 4270–4275 (2010).
[7] Perception-based problem solver (PbPS): an open-source MATLAB toolbox, link for download: http://web2.um.ac.ir/~akbazar/uploads/Main/PbPS.zip.
[8] Pedrycz, W. "The design of cognitive maps: a study in synergy of granular computing and evolutionary optimization", *Expert Systems with Applications*, 37(10), pp. 7288–7294 (2010).
[9] Tao, Y.Q., et al. "Knowledge evolutionary algorithm based on granular computing", *2008 IEEE Conference on Cybernetics and Intelligent Systems*, pp. 341–346 (2008).
[10] Batyrshin, I. and Sheremetov, L. "Perception-based approach to time series data mining", *Applied Soft Computing*, 8(3), pp. 1211–1221 (2008).
[11] Qin, Z., et al. "Deduction engine design for PNL-based question answering system", In *Foundations of Fuzzy Logic and Soft Computing*, In *Lecture Notes in Computer Science*, vol. 4529/2007, pp. 253–262, Springer (2007).
[12] Gacek, A. and Pedrycz, W. "A granular description of ECG signals", *IEEE Transactions on Biomedical Engineering*, 53(10), pp. 1972–1982 (2006).
[13] Reformat, M. and Pedrycz, W. "Evolutionary optimization of information granules", *2001 IFSA World Congress and 20th NAFIPS International Conference*, 4, pp. 2035–2040 (2001).
[14] Ho, Y.C. and Lee, J.T. "Granular optimization: An approach to function optimization", *Proceedings of the 39th IEEE Conference on Decision and Control*, 1, pp. 103–111 (2000).
[15] Zadeh, L.A. "From computing with numbers to computing with words", *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, 45(1), pp. 105–119 (1999).
[16] Zadeh, L.A. "Fuzzy logic = computing with words", *IEEE Transaction on Fuzzy Systems*, 4(2), pp. 103–111 (1996).
[17] Zadeh, L.A. "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic", *Fuzzy Sets and Systems*, 90, pp. 111–127 (1997).
[18] Zadeh, L.A. "A new direction in AI, toward a computational theory of perceptions", *AI Magazine*, pp. 73–84 Spring 2001 (2001).
[19] Zadeh, L.A. "Precisiated natural language, toward a radical enlargement of the role of natural languages in information processing, decision and control", *Proceedings of the 9th International Conference on Neural Information Processing*, 1, pp. 1–3 (2002).
[20] Zadeh, L.A. "Precisiated natural language, PNL", *AI Magazine*, 25(3), pp. 74–91 (2004).
[21] Zadeh, L.A. "Granular computing — the concept of generalized constraint-based computation", In *Rough Sets and Current Trends in Computing*, In *Lecture Notes in Computer Science*, vol. 4259/2006, pp. 12–14, Springer (2006).
[22] Zadeh, L.A. "Generalized theory of uncertainty (GTU), principal concepts and ideas", *Computational Statistics & Data Analysis*, 51, pp. 15–46 (2006).
[23] Zadeh, L.A. "Toward human level machine intelligence, is it achievable? The need for a paradigm shift", *IEEE Computational Intelligence Magazine*, pp. 11–22 (2008).

Alireza Rowhanimanesh is currently a Doctoral Student of Control Engineering at Ferdowsi University of Mashhad (FUM). His Ph.D. Thesis is on Perception-based Analysis and Control of Complex Systems under the supervision of Prof. M.-R. Akbarzadeh-T. He received his B.S. and M.S. Degrees in Control Engineering from FUM in 2007 and 2009, respectively. He has received several awards including Outstanding Graduate Student Award in 2009, Best Paper Award in ISFS2008 and Outstanding Student Awards in 2007 and 2003. His research interests are in the areas of Analysis and Control of Biological Systems, Perception-Based Computing, Complex Systems, Soft Computing and Robotics. He has published over 25 peer-reviewed articles in these and related research fields.

Mohammad-R. Akbarzadeh-T is currently Professor of Electrical Engineering and Computer Engineering at Ferdowsi University of Mashhad. In 1999, he received his Ph.D. in Evolutionary Optimization and Fuzzy Control of Complex Systems from the University of New Mexico. In 2006–2007, he completed a one year visiting scholar position at Berkeley Initiative on Soft Computing (BISC), UC Berkeley. Dr. Akbarzadeh has received several awards including the IDB Excellent Performance and Leadership Awards in 2009 and 2010, and the Outstanding Faculty Awards in 2008 and 2002. His research interests are in the areas of Evolutionary Algorithms, Fuzzy Logic and Control, Soft Computing, Multi-Agent Systems, Complex Systems, Robotics and Biomedical Engineering Systems. He has published over 230 peer-reviewed articles in these and related research fields.