Evolutionary Variable Identification

Antonio F. Gómez-Skarmeta, Fernando Jiménez, Jesús Ibáñez, Santiago Paredes
Dept. Informática, Inteligencia Artificial y Electrónica
Universidad de Murcia
Campus de Espinardo, 30071-Espinardo, Murcia, Spain
Phone: +34-68-364630, Fax: +34-68-364151
e-mail: {skarmeta,fernan,jesus,chapu}@dif.um.es

ABSTRACT: Variable identification problem which usually appears in Knowledge Discovery, Data Mining or Qualitative Modeling processes, is solved in this paper from a fuzzy-evolutionary perspective. Fuzzy Logic is showed as appropiated to approach the problem when there is uncertainty concerning the reliability of the observed data, and Evolutionary Algorithms appear as powerfull techniques of global optimization capables to find better solution in comparison with other optimization techniques as gradient methods.

KEYWORDS: Variable identification; knowledge discovery; data mining; qualitative modeling; fuzzy logic; evolutionary algorithms.

1 INTRODUCTION

Preprocessing of the data in order to select the most adequate variables from a set of variable candidates is usually called *variable identification.* This problem appears in important areas such as *Knowledge Discovery*, *Data Mining* and *Qualitative Modeling.* As in (Fayyad, 1996) we use the term Knowledge Discovery in Databases (KDD) or just Knowledge Discovery (DM) to denote the overall process of extracting high-level knowledge from low-level data, and the term Data Mining as the concrete act of extracting patterns or model from huge records of data. Hence many steps precede data mining and one of them is the preprocessing of the data in order to select the adequate variables to be used in the model identification. This step is similar to the variable identification within the structure identification step in the qualitative modeling process (Sugeno and Yasukawa, 1993).

Variable identification problem can be formulated as a constrained optimization problem, i.e. as a *Mathematical Programming* one. Thus, given a collection $S = \{(x_1^1, \ldots, x_p^1, y_1), \ldots, (x_1^n, \ldots, x_p^n, y_n)\}$ of normalized data, i.e. n observed data each one of them composed of p input variables and one output variable (we assume a Multiple Input Single Output system without loss of generality) for which some functional dependency is presumed, one must determine the weight related with each variable such that the *mean quadratic error* is minimized. The idea here is that initially we have a collection of variables that, maybe not all of them, have a similar grade of influence on the functional dependency or either they are noisy variables. Hence each of the p weigths can have any value within the interval $[0, 1]$, and the weights sum up to one. The mathematical expression can be as follows:

$$\begin{array}{ll} Minimize & E = \sqrt{\dfrac{\sum_{t=1}^{n}\left(y_t - \sum_{j=1}^{p} w_j x_j^t\right)^2}{n}} \\ s.t.: & \\ & \sum_{j=1}^{p} w_j = 1,\ 0 \le w_j \le 1,\ j = 1, \ldots, p \end{array} \tag{1}$$

On this way, those variables with a related weight sufficiently small in the optimal solution can be ignored from the input variable set which will be present in the final model.

Nevertheless, as we attempt to solve real-world problems, we realize that they are tipically ill-defined systems, and usually there is uncertainty concerning the reliability of the observed sample. In these cases, we need approximate reasoning approaches capable of handling such imperfect information. *Soft Computing* (Bonissone,

```
procedure EA
begin
    initialize_population;
    evaluate_population;
    while (not termination-condition) do
    begin
        generate_new_population; {selection, replication, variation
                                  and generational replacement}
        evaluate_population;
    end
end
```

Figure 1: Structure of an Evolutionary Algorithm.

1997) is a recently coined term describing the symbiotic use of many emerging computing disciplines that try to handle the imperfect information. In contrast to traditional (hard computing), soft computing is tolerant of imprecision, uncertainty, and partial truth (Delgado et al., 1997; Vila et al., 1997). Within this techniques we have *Fuzzy Logic*. From the perspective of the Fuzzy Logic, and using the *generalized mean* model (Dyckhoff and Pedrycz, 1984), the variable identification problem can be formulated as follows:

$$\begin{array}{ll} Minimize & E = \sqrt{\dfrac{\sum_{t=1}^{n}\left(y_t - \left(\sum_{j=1}^{p} w_j {x_j^t}^f\right)^{1/f}\right)^2}{n}} \\ s.t.: & \\ & \sum_{j=1}^{p} w_j = 1,\ 0 \le w_j \le 1,\ j = 1,\ldots,p, \\ & f_l \le f \le f_u \end{array} \tag{2}$$

where $f \in [f_l, f_u]$ $(-\infty \le f_l \le f_u \le +\infty)$ is the degree of fuzziness which expresses the degree of redundancy/complementarity of the observed data. This kind of aggregation operator is similar to the OWA (*Ordered Weighted Averaging*) operator introduced by Yager (1988). Although numerous fuzzy sets connectives can be used for the purpose of aggregation (Zimmermann, 1987), desirable properties are satisfied by the generalizated mean operator. The degree of fuzziness f can be either provided by a decision maker or required by the problem and then must to be calculated. Note that in formulation (2) the degree of fuzziness f can be established to a concrete value $\overline{f}$ setting $f_l = \overline{f} = f_u$.

To solve this problem, the first question is to determine the more appropiate technique according to the kind of the problem. The problem (2) is a nonlinear problem with linear constraints and numerous techniques have been described for this proposal, e.g. *gradient methods* (Wolfe, 1966), although these aim at local optimization only. *Evolutionary Algorithms* (EA) (Biethahn and Nissen, 1995; Goldberg, 1989; Michalewicz, 1992) have been shown in the last years as more appropiate for this kind of problems since they are global optimization methods that aim at complex objective functions and constraints (Gómez-Skarmeta and Jiménez, 1997).

With this background, we are interested in the analysis of evolutionary computation techniques to solve the variable identification problem as in (2). Thus, the paper is organized as follows: Section 2 describes the structure and components of an *ad hoc* EA to solve the considered problem. Simulation results and comparison with other non evolutionary optimization technique are shown in section 3. Finally, section 4 indicates the main conclusions and future works.

2 AN EVOLUTIONARY ALGORITHM FOR VARIABLE IDENTIFICATION

The problem (2) can be solved by using evolutionary techniques from two different perspectives: 1) by means of an ad hoc EA specialized on the problem, and 2) with an evolutionary algorithm for general proposal. In both approximations the structure of the EA can be as figure 1 shows. In this paper we describe an evolutionary approximation of the first type (ad hoc EA) since it seems (Michalewicz, 1992) that a natural representation of a potential solution for a given problem, together with a family of applicable variation operators might be quite useful in the approximation of solutions to many problems.

To design an ad hoc EA to solve the problem is important to realize that, on the one hand, we can representate on a natural way the solutions of the problem with chromosomes composed by $p+1$ float point values (p weights and the degree of fuzziness f). On the other hand, it is easy to design efficient initialization procedures and variation operators such that all constraints in the problem are satisfied by the new individuals.

Buczak and Uhrig (1996) describe a Genetic Algorithm (bit-string representation) to solve the same problem with application to data fusion. However these authors do not take into account above considerations and repair algorithms are used after crossover and mutation to satisfy the constraints. The representation of the solutions and the mechanism of constraints satisfaction are in this way the main differences w.r.t. the EA described in this paper.

In the following, the main characteristics of the EA are briefly described. These characteristics are a representation of solutions to the problem, handled constraints, mechanisms to create an initial population of solutions, an evaluation function, variation operators and parameters used. Moreover, some remarks on design such as selection and generational replacement mechanisms are made.

2.1 REPRESENTATION

An individual V of the population is represented as a pair $V = (W, f)$ where W is an array $W = (w_1, \dots, w_p)$ with p float point components (w_j, $j = 1, \dots, p$, stand for the weight related with the input variable x_j), and f is a float point value (standing for the degree of fuzziness).

2.2 CONSTRAINT SATISFACTION

All mechanisms to create a new individual $V = (W, f)$ in the evolutionary process, i.e. initialization procedures and variation operators, ensure that the constraints $\sum_{j=1}^{p} w_j = 1$, $0 \le w_j \le 1$, $j = 1, \dots, p$, $f_l \le f \le f_u$, are satisfied.

2.3 INITIAL POPULATION

The following procedure *initialize_population* obtains a population $POP = (V_1, \dots, V_{popsize})$ of *popsize* individuals, with $V_i = (W_i, f_i)$, $W_i = (w_1^i, \dots, w_p^i)$, which satisfies the imposed constraints. We consider two initialization procedures. The procedure *weights1* is a general one that receiving a number q and a value val, it generates an array of weights $W = (w_1, \dots, w_q)$ such that $w_l \in [0, 1]$, $l = 1, \dots, q$, and $\sum_{l=1}^{q} w_l = val$, with $0 \le val \le 1$. The procedure *weights2* is a modification of the procedure weights1 to set weights equal to zero in a random ratio. Note that, for $q = p$ and $val = 1$, both procedures weights1 and weights2 generate a feasible solution for the problem. These procedures are used with equal probability to obtain the initial population. The degree of fuzziness is separately generated at random from the domain $[f_l, f_u]$.

```
procedure initial_population;
begin
    $i \leftarrow 1$;
    while $i \le popsize$ do
    begin
        call $\left\{ \begin{array}{c} \text{weights1}(input: p, 1; output: W) \\ \text{or} \\ \text{weights2}(input: p, 1; output: W) \end{array} \right\}$ with equal probability;
        $f \leftarrow$ random real value $\in [f_l, f_u]$;
        $V_i \leftarrow (W, f)$;
        $i \leftarrow i + 1$;
    end
end
procedure weights1;
input: integer number $q$, with $1 \le q \le p$; real value $val$, with $0 \le val \le 1$; output: array $W = (w_1, \dots, w_q)$ such that $w_l \in [0, 1]$, $l = 1, \dots, q$, and $\sum_{l=1}^{q} w_l = val$;
begin
    $w_l \leftarrow$ random real value $\in (0, 1]$, for $l = 1, \dots, q$;
```

```
    $w_l \leftarrow val \cdot w_l / \sum_{i=1}^{q} w_i$, for $l = 1, \dots, q$;
end
procedure weights2;
input: integer number $q$, with $1 \le q \le p$; real value $val$, with $0 \le val \le 1$; output: array $W = (w_1, \dots, w_q)$ such that $w_l \in [0,1]$, $l = 1, \dots, q$, and $\sum_{l=1}^{q} w_l = val$;
begin
    set randomly $K = \{k_1, \dots, k_r\} \subseteq \{1, \dots, q\}$ such that $1 \le r \le q$;
    $M \leftarrow \{1, \dots, q\} - K$;
    $w_l \leftarrow 0$, $l \in M$;
    weights1($input : r, val; output : V$);
    $w_{k_l} \leftarrow v_l$, $l = 1, \dots, r$;
end
```

2.4 EVALUATION FUNCTION

Evaluation function of the individuals $V_i = (W_i, f_i)$, with $W_i = (w_1^i, \dots, w_p^i)$, $i = 1, \dots, popsize$, is clearly determined by the objective function of the problem:

$$eval(V_i) = \sqrt{\frac{\sum_{t=1}^{n} \left(y_t - \left(\sum_{j=1}^{p} w_j^i {x_j^t}^{f_i} \right)^{1/f_i} \right)^2}{n}}$$

2.5 SELECTION MECHANISM AND GENERATIONAL REPLACEMENT

We propose to use the *tournament selection* (Goldberg, 1989). In this method (see figure 2), a group of n_{tourn} individuals is sampled from the population and the individual with the best fitness in the group is chosen for reproduction. Variation operators are applied to the selected individuals and the offspring are copied to the next population. This process is repeated until the whole new population is generated (*complete generational replacement*). Moreover *elitism strategy*, which always copies the best member of a population to the next population, is used. Note that replication of individuals is achieved when no variation operator is applied. The complete procedure to generate a new population is shown in figure 3. In generate_new_population procedure, POP and NEW_POP represent the current population and the new population respectively, and NEW_POP(s).IND represents the individual which is placed at the position s in the new population.

```
function tournament_selection
begin
    Set randomly $J = \{j_1, \dots, j_{tourn}\} \subseteq \{1, \dots, popsize\}$
    return(best($J$))
end
```

Figure 2: Tournament selection.

2.6 VARIATION OPERATORS

We have considered four variation operators (arithmetical crossover and three different mutations). The operator *arithmetical_crossover* produces two offsprings by means of convex linear combinations of the parents. All mutation operators work separately with the components W_i and f_i of an individual V_i. Thus, the operator *mutation1* makes a minimal change in the components of W_i of the parent, swapping two random elements, whereas the operators *mutation2* and *mutation3* use the procedures weight1 and weight2 respectively to produce a change in an arbitrary subarray from the components of W_i of the parent. The component f_i is always mutated via non-uniformity mutation (Michalewicz, 1992) and then a new value f_i' is obtained as follows:

$$f_i' = \begin{cases} f_i + (f_u - f_i) \cdot r \cdot (1 - \frac{t}{T})^c, & \text{if a random digit is 0} \\ f_i - (f_i - f_l) \cdot r \cdot (1 - \frac{t}{T})^c, & \text{if a random digit is 1} \end{cases}$$

```
procedure generate_new_population
begin
    $s \leftarrow 1$;
    $I \leftarrow \{1, \ldots, popsize\}$;
    NEW_POP($s$).IND $\leftarrow$ best($input : I$); {elitism strategy}
    while $s < popsize$ do
    begin
        $mate1 \leftarrow$ tournament_selection;
        $mate2 \leftarrow$ tournament_selection;
        arithmetical_crossover($input : mate1, mate2; output : child1, child2$);
        $offspring1 \leftarrow \left\{ \begin{array}{c} \text{mutation1}(input : child1) \\ \text{or} \\ \text{mutation2}(input : child1) \\ \text{or} \\ \text{mutation3}(input : child1) \end{array} \right\}$ with equal probability;
        $offspring2 \leftarrow \left\{ \begin{array}{c} \text{mutation1}(input : child2) \\ \text{or} \\ \text{mutation2}(input : child2) \\ \text{or} \\ \text{mutation3}(input : child2) \end{array} \right\}$ with equal probability;
        $s \leftarrow s + 1$;
        NEW_POP($s$).IND $\leftarrow offspring1$;
        $s \leftarrow s + 1$;
        if $s \leq popsize$ then
            NEW_POP($s$).IND $\leftarrow offspring2$;
    end
    POP $\leftarrow$ NEW_POP;
end
```

Figure 3: A procedure to obtain a new population.

where $[f_l, f_u]$ is the domain of the variable f_i, r is a random number from $[0, 1]$, T is the maximal generation number, t is the present generation, and c is a system parameter determining the degree of non-uniformity.

procedure arithmetical_crossover;
input: Parents $V_1 = (W_1, f_1)$, with $W_1 = (w_1^1, \ldots, w_p^1)$, and $V_2 = (W_2, f_2)$, with $W_2 = (w_1^2, \ldots, w_p^2)$; **output:** Children $V_1' = (W_1', f_1')$, with $W_1' = ({w'}_1^1, \ldots, {w'}_p^1)$, and $V_2' = (W_2', f_2')$, with $W_2' = ({w'}_1^2, \ldots, {w'}_p^2)$;

```
begin
    $rnd \leftarrow$ random value $\in [0, 1]$;
    if $rnd \leq p_c$ then
    begin
        $c_1 \leftarrow$ random real value $\in [0, 1]$; $c_2 \leftarrow 1 - c_1$;
        ${w'}_j^1 \leftarrow c_1 \cdot w_j^1 + c_2 \cdot w_j^2$, $j = 1, \ldots, p$; ${w'}_j^2 \leftarrow c_2 \cdot w_j^1 + c_1 \cdot w_j^2$, $j = 1, \ldots, p$;
        $f_1' \leftarrow c_1 \cdot f_1 + c_2 \cdot f_2$; $f_2' \leftarrow c_2 \cdot f_1 + c_1 \cdot f_2$;
    end
    else
    begin
        $V_1' \leftarrow V_1$; $V_2 \leftarrow V_2$;
    end
end
```

procedure mutation1;
input: Parent $V = (W, f)$, with $W = \{w_1, \ldots, w_p\}$; **output:** Offspring $V' = (W', f')$, with $W' = \{w_1', \ldots, w_p'\}$;

```
begin
    $rnd \leftarrow$ random value $\in [0, 1]$;
    if $rnd \leq p_{m1}$ then
    begin
        set randomly $K = \{k_1, k_2\} \subseteq \{1, \ldots, p\}$;
        $M \leftarrow \{1, \ldots, p\} - K$;
        $w_l' \leftarrow w_l$, $l \in M$;
        $w_{k_1}' \leftarrow w_{k_2}$;
        $w_{k_2}' \leftarrow w_{k_1}$;
        $f' \leftarrow$ non_uniform_mutation($input : f$);
    end
    else $V' \leftarrow V$;
```

```
end
procedure mutation2;
input: Parent $V = (W, f)$, with $W = \{w_1, \dots, w_p\}$; output: Offspring $V' = (W', f')$, with $W' = \{w'_1, \dots, w'_p\}$;
begin
    $rnd \leftarrow$ random value $\in [0, 1]$;
    if $rnd \leq p_{m2}$ then
    begin
        set randomly $K = \{k_1, \dots, k_r\} \subseteq \{1, \dots, p\}$ such that $1 < r \leq p$;
        $M \leftarrow \{1, \dots, p\} - K$;
        $w'_l \leftarrow w_l,\ l \in M$;
        $val \leftarrow \sum_{l=1}^{r} w_{k_l}$;
        weights1($input : r, val; output : V$);
        $w'_{k_l} \leftarrow v_l,\ l = 1, \dots, r$;
        $f' \leftarrow$ non_uniform_mutation($input : f$);
    end
    else $V' \leftarrow V$;
end
procedure mutation3;
input: Parent $V = (W, f)$, with $W = \{w_1, \dots, w_p\}$; output: Offspring $V' = (W', f')$, with $W' = \{w'_1, \dots, w'_p\}$;
begin
    $rnd \leftarrow$ random value $\in [0, 1]$;
    if $rnd \leq p_{m3}$ then
    begin
        set randomly $K = \{k_1, \dots, k_r\} \subseteq \{1, \dots, p\}$ such that $1 < r \leq p$;
        set $M = \{1, \dots, p\} - K$;
        $w'_l \leftarrow w_l,\ l \in M$;
        $val \leftarrow \sum_{l=1}^{r} w_{k_l}$;
        weights2($input : r, val; output : V$);
        $w'_{k_l} \leftarrow v_l,\ l = 1, \dots, r$;
        $f' \leftarrow$ non_uniform_mutation($input : f$);
    end
    else $V' \leftarrow V$;
end
```

2.7 PARAMETERS

The EA parameter values for which we have obtained good results in the experimentation process are $maxgen = 1000$ (maximal generation number), $popsize = 20$ (population size), $n_{tourn} = 6$ (number of members used in the tournament selection), $p_c = 0.6$ (probability of arithmetical crossover), $p_{m1} = 0.6$ (probability of mutation1), $p_{m2} = 0.6$ (probability of mutation2), $p_{m3} = 0.6$ (probability of mutation3), and $c = 2$ (degree of non-uniformity).

3 EXPERIMENTS AND RESULTS

To validate the EA we have considered a standard test proposed by Dickhoff and Pedrycz (1984) and used in Buczak and Uhrig (1996). The data set consists of a collection of two input variables x_1 and x_2, and an output variable. To check the performance of the EA, we have added six new variables x_3, x_4, x_5, x_6, x_7 and x_8, constructed from x_1 and x_2 by adding 10% (to x_3 and x_4), 40% (to x_5 and x_6), and 100% (to x_7 and x_8) uniform noise. Table 1 shows the results obtained considering only the original input variables, for 2 different values of fuzziness f ($f = 1$ and $f = 2$, setting by a decisor), and when the optimal fuzziness f^* has to be calculated (within two different intervals $[f_l, f_u] = [-10, 10]$ and $[f_l, f_u] = [-20, 20]$). Table 2 shows the results obtained considering the original input variables and the added fictitious noisy variables. The results show that the EA determines the variables which really have influence over the output variable, excluding those fictitious introduced artificially. For the sake of comparison with other non evolutionary techniques such as gradient methods (Grace, 1990) we also show in Table 3 the results (fitness) obtained (on average over ten runs using different random number starting seeds for the EA and different random starting points for the gradient method) for the two above test problems with eight variables in which the fuzziness f has to be calculated within the intervals $[f_l, f_u] = [-10, 10]$ and $[f_l, f_u] = [-20, 20]$. As we can appreciate the EA obtains better results than the gradient method with improvement of 70.68% on average over the two test problems.

Table 1: EA results considering just the sample.

	$f = 1$	$f = 2$	$f^* = 0.381981$ $[f_l, f_u] = [-10, 10]$	$f^* = 0.382001$ $[f_l, f_u] = [-20, 20]$
w_1	0.423380	0.315553	0.447067	0.447028
w_2	0.576620	0.684447	0.552933	0.552972
fitness	0.092584	0.160683	0.052519	0.052519

Table 2: EA results considering the sample with noise.

	$f = 1$	$f = 2$	$f^* = 0.397632$ $[f_l, f_u] = [-10, 10]$	$f^* = 0.382001$ $[f_l, f_u] = [-20, 20]$
w_1	0.383144	0.314625	0.359739	0.414895
w_2	0.567877	0.614990	0.484674	0.513846
w_3	0.000000	0.000000	0.071508	0.004978
w_4	0.000000	0.070384	0.008297	0.010950
w_5	0.031664	0.000000	0.008303	0.000000
w_6	0.000000	0.000000	0.004109	0.016615
w_7	0.017315	0.000000	0.013623	0.020711
w_8	0.000000	0.000000	0.049747	0.018006
fitness	0.091591	0.160602	0.052082	0.051806

Table 3: Comparison between gradient method and EA.

	Gradient method	EA	Difference
$[f_l, f_u] = [-10, 10]$	0.176019	0.054576	−68.99%
$[f_l, f_u] = [-20, 20]$	0.205620	0.056829	−72.36%

4 CONCLUSIONS

The variable identification problem appears in different research areas as Data Mining, or Qualitative Modeling, hence it is an important task in many aspects that justify the attention given to it in this paper. Due the natural uncertainty the observed data have, we have approched the problem within the context of Fuzzy Logic. In this paper we show an ad-hoc EA to solve the non linear optimization problem that underlies the variable identification problem as we have defined it. The results obtained show that the EA is able to find better solutions than classical non evolutionary techniques as gradient ones. As a future trend, although our EA have shown good behaviour, in order to apply it to real Data Mining problems where we must work with huge amount of data, this particular EA could not be appropiated. We are evaluating different alternatives in order to integrated within the EA, techniques for the manipulation of large volumes of data.

ACKNOWLEDGMENTS

We would like to thank the *Comisión Interministerial de Ciencia y Tecnología* (*CICYT*) for its support through the projects TIC97-1343-C02-02 and 1FD97-0255-C03-01. As well, we thank the *Instituto de Fomento de la Región de Murcia* for its financing through the Seneca Program.

REFERENCES

Biethahn, J., Nissen, V. (Eds) (1995). *Evolutionary Algorithms in Management Applications.* Springer-Verlag Berlin Heidelberg.

Bonissone, P. (1997). “Soft Computing: The Convergence of Emerging Reasoning Techniques”, Journal of Soft Computing, vol. 1, n. 1.

Buczak, A.L., Uhrig, R.E. (1996). “Hybrid Fuzzy-Genetic Technique for Multisensor Fusion”, Information Sciences, vol. 93, pp. 265-281.

Delgado, M., Gómez Skarmeta, A., Martín, F. (1997). “Some Methods to Model Fuzzy Systems for Inference Purposes”, Int. Journal of Approximate Reasoning, vol. 16, n. 3-4, pp. 377-391.

Dyckhoff, H., Pedrycz, W. (1984). “Generalized Means of Model of Compensative Connectives”, Fuzzy Sets and Systems, vol. 14, pp. 143-154.

Fayyad, U. (1996). “Data Mining and Knowledge Discovery: Making Sense Out of Data”, IEEE Expert, pp. 20-25.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley.

Gómez-Skarmeta, A., Jiménez, F. (1997). “Generating and Tuning Fuzzy Rules using Hybrid Systems”, Proceeding of the FUZZ/IEEE97, pp. 247-252, Barcelona, Spain.

Grace, A. (1990). *Optimization Toolbox for Use with MATLAB.* The MathWorks, Inc..

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs.* Springer Verlag.

Sugeno, M., Yasukawa, T. (1993). “A Fuzzy-Logic-Based Approach to Qualitative Modeling”, IEEE Trans. on Fuzzy Systems, vol.1, n. 1, pp. 7-31.

Vila, M.A., Delgado, M., Gómez Skarmeta, A.F. (1997). “On the Use of Probability and Possibility Measures in Fuzzy Clustering”, Proceeding of the FUZZ/IEEE97, pp. 143-148, Barcelona, Spain.

Wolfe, P. (1966). “On the Convergence of Gradient Methods Under Constraints”, IBM Research Report RZ 204, Zurich, Switzerland.

Yager, R.R. (1988). “On Ordered Weighted Averaging Aggregation Operators in Multi-Criteria Decision Making”, IEEE Transactions on Systems, Man and Cybernetics, vol. 18, pp. 183-190.

Zimmermann, H.J. (1987). *Fuzzy Sets, Decision Making and Expert Systems.* Kluwer Academic.