

A HEURISTIC ALGORITHM THAT FINDS AN OPTIMUM SOLUTION FOR A MAMDANI FUZZY INFERENCE SYSTEM

Ovidiu Turcoane

Bucharest University of Economic Studies - Cybernetics, Statistics and Economic Informatics
turcoaneovidu06@stud.ase.ro

ABSTRACT

Most of the techniques of nonlinear optimization assume the existence of a differential function, but there also exist other approaches. In this paper, we propose a non-differential algorithm that solves the problem of minimization and maximization for any objective function of several unknown variables. The algorithm searches for a solution vector of variables that yields the optimal result of the objective function, using a trial-and-error process. The algorithm is not fast, but it converges to a non-trivial solution, if it exists. The method will be used to determine the minimum and the maximum of the outputs of a sociopolitical model of E-Democracy. This model of E-Democracy is based on a Mamdani fuzzy inference system. We provide formalization and Matlab implementation for our fuzzy system and for the algorithm.

Keywords

Nonlinear optimization, non-differential optimization algorithm, Mamdani inference system, E-Democracy model, direct search.

1. INTRODUCTION

Generally, the matter of optimization belongs specifically to human nature and it may be the key of the evolution of mankind. There has been a continuous struggle for the human race to ameliorate its condition, and things seem to keep the same trend, at least for short and medium term.

It is a natural consequence that a multitude of solutions have been proposed for plenty of types of optimization problems. These include methods of optimized search for better economic, social or mathematical solutions, as well as a continuous search to improve the methods themselves, or even discover new ones.

From a mathematical (and quite a from a human reasoning) point of view the problem of optimization consists of:

- several unknown variables
- an objective function depending on the unknown variables, modeling the purpose of the problem
- a system of equalities and inequalities, the constraints of the problem

The general case is when the objective function or the constraints are mathematically formalized as nonlinear functions, and this may normally be the case of non-trivial human reasoning. The goal of the optimization process is to find a maximum or a minimum for the objective function. Formula (1) describes the generic optimization problem:

$$\begin{cases} \max_{x \in X} f(x) \text{ or } \min_{x \in X} f(x) \\ h_i(x) = 0, i = 1, \dots, m \\ g_j(x) \leq 0, j = 1, \dots, p \end{cases}$$

where

x is a subset/vector of n unknown variables, $x \in X$ (1)

X is the set/domain of instances of x , $X \subset R^n$

f is the objective function, $f: R^n \rightarrow R$

h, g are the constraint functions, $h, g: R^n \rightarrow R$

m, p are the number of equalities and, respectively, inequalities of constraints

We are interested in discussing the case where the problem has only one objective function f and this function is not differentiable. In specific literature, we identify several non-differential optimization algorithms. Some of them assume a sort of partial differentiability or sub-gradient approach: generic cutting plane algorithm, minimax problems, quasi-Newton method, parametric programming, relaxation methods, sub-gradient optimization [1]. Yet, we are interested in solutions that do not require any kind of differentiability for f , and these are better known as direct search methods. While in the 1970's direct search approaches had been dismissed by the proponents of differential optimization for three reasons: heuristic development, no proof of convergence, and slow convergence rate, the interest for them has lately increased for two reasons: single available solution and proof of convergence for large number of direct search methods [2]. Those who coined the term of direct search [3] pointed out on their simplicity and adaptedness to electronic computations. From the practical side, three broad categories of direct search methods are identified [4]:

- pattern search methods: procedures that search for best improvement of f each time a parameter varies according to predetermined steps;
- simplex methods (distinctive from homonymous linear programming method): creating a $n + 1$ set of points (*i.e.* a simplex): the procedure searches for an improvement of f reflecting the worst vertex in the simplex through the centroid of the opposite face (*i.e.* centroid of the remaining vertices);
- methods with adaptive sets of search directions: procedures that adapt the search to the f curvature by using the information available in the previous steps.

Each of these three broad categories employs several methods, different approaches of non-derivative direct search methods being necessary since there is no such thing as *one method fits all*. In this paper, we propose an algorithm that is not intended to be original, but to be particularly effective for achieving the optimization of the Mamdani fuzzy inference system.

Mamdani fuzzy inference system (MFIS) is among the first fuzzy methodologies to be dealt with, and it continues to be a very common one. It uses linguistic control rules to model decision or complex systems problems and it is very close related to human reasoning [5], which also makes it very popular. However, other types of fuzzy inference (*e.g.* where the objective function has mathematical crisp formalization) are sometimes preferred, since algorithms which can be used to automatically optimize them already exist. The objective function f of an MFIS is determined through a process of several transformations that cannot be simply formalized mathematically in one single expression. Yet, we propose a method to optimize it through a direct search approach that will be further discussed in this paper.

In a first attempt, we presented a model of E-Democracy that is based on an MFIS with five inputs and one output [6]. We introduce in this paper a simplified MFIS approach of this model, with only three inputs and one output. We briefly discuss this new model of E-Democracy (NME), providing only the basics of the MFIS that supports it. Extensive discussions on

NME will be subject to further research. However, this paper addresses the optimization in an MFIS, so the NME is going to be used only for the practical reason of introducing our algorithm for optimum E-Democracy (AOE).

This article is organized as follows: section 2 briefly describes the logics behind MFIS in the context of NME and builds the specific problem using a Matlab approach; AOE is discussed in section 3; in section 4, some comparisons are made between AOE and methods for a differential optimization problem; the last section concludes this paper.

2. E-DEMOCRACY MODEL AND ITS MFIS

There is plenty of printed material on MFIS, but any discussion in this direction is considered to be unproductive for our article purpose. Nevertheless, we remind the components of MFIS [7] and we also define, using Matlab functions, the fuzzy logic (FL) of our model by specifying methods used for each component:

- i. *Fuzzification* – building the fuzzy subsets of the inputs, defined by membership functions, using crisp input values
- ii. *Definition of knowledge base* – applying fuzzy operators by building rules of MFIS: *min* for AND, *max* for OR
- iii. *Implication method* – determining an intermediary output (*O*) for each rule, based on a single or multiple fuzzy subsets: *prod*
- iv. *Aggregation of all outputs* – combining every intermediary output *O* of each rule to create a single final output (*F*): *sum*
- v. *Defuzzification* – extracting the crisp value from final output *F*: *centroid*

FL is not important when discussing AOE, as the latter applies to any kind of logic or fuzzy inference system, but it is necessary to better frame NME. We end our short generic presentation of an MFIS with an illustration in Figure 1.

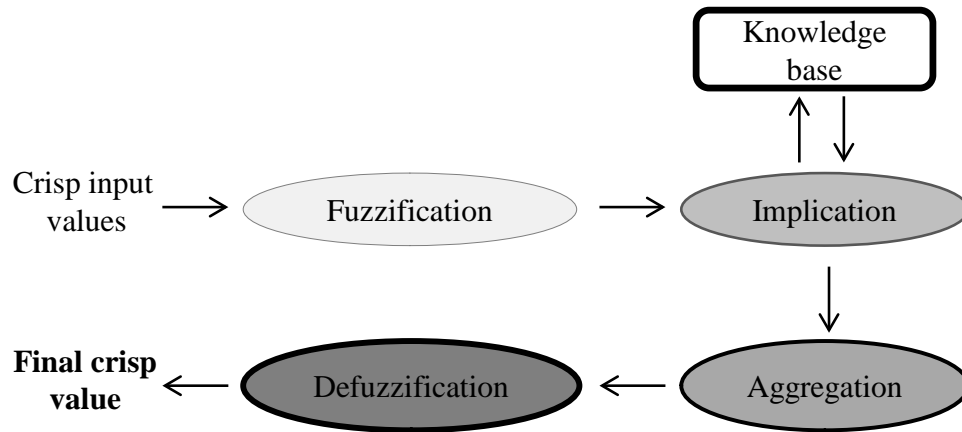


Fig 1: MFIS components

Before introducing our MFIS for NME, let us briefly discuss the rationale behind NME. E-democracy has been seen as an extension of E-Government [8], but this is not the case. In order to define E-democracy, a brief description of the knowledge society is required, since this represents the very foundation upon which it is built. The term “knowledge society” stands for a large community where people are dependent on each other and members are considered to be equally important, since one’s work and experience constitute a strong contribution to achieving individual and common ends. Knowledge is built on the concept of information, which, in a global society, must be prevented from being impeded, manipulated and/or censored. Furthermore, as palpable and abstract tools of knowledge society, information and communication technologies have the purpose of identifying, producing, processing, transforming, disseminating and using information for the benefit of human development through knowledge [9]. The continuous cooperation between members of knowledge society is based on non-sum-zero games, and the attempt of one member to deceive another leads to poorer individual and common results [10].

The main political institutions of representative democracy or of the state subject to the rule of law are: parliament, executive (govern and presidency/constitutional monarchy) and justice. We build our model of E-democracy on these institutions, but we make a substantial improvement by adding another institution: citizenry. Citizenry is not just a body that includes all of the citizens, but it is an expression of general will (*volonté générale*) and common interest [11], without neglecting the freedom of individuals [12]. Justice is *the* institution that prevents citizens or representatives from imposing their tyranny [13]. It is also the watcher and preserver of the rights of each community, minority or individuals and is the key for a society of inclusion, no matter of religion, gender, race, age, beliefs etc. [14].

Thus, the three inputs of E-democracy’s model based on FLs and MFIS are Citizenry, Justice, and Delegates and they yield the E-democracy output. Firstly, we define our model of E-democracy by establishing the membership functions (MFs) of the fuzzy subsets that build the fuzzy sets (FSs) of the three inputs and one output. Secondly, we build the rules of E-democracy (REDs) using natural language and afterwards we provide a mathematical formalization of these rules. All mathematical and computational formulae are presented herein with respect to the established Matlab formalization [15]. For functions and methods used in this research, which are not already implemented in Matlab, an algorithm and/or a source code will be provided.

In addition, the following notations are going to be introduced: type of each MF (TMF), name of the fuzzy subset (NMF) and values defined (VD) of each MF. VD has five components: deviation or uncertainty (σ); lower bound (lb) and upper bound (ub) that delimit the crisp interval of each MF; and the approximate left value (lv) and approximate right value (rv) of each MF support [16].

MFs of all FSs are presented in Table 1: FS of Citizenry (FC), FS of Justice (FJ), FS of Delegates (FD) and FS of E-democracy (FE) and each MF is described by Matlab function *gauss2mf* [15], using a 0-1 scale for inputs and output.

Table 1. MFs of E-democracy model

FS	NMF	TMF	VMF	VD				
				σ	lv	lb	ub	rv
FC	non	<i>gauss2mf</i>	(1.0000 0.000 0.065 0.200)	0.065	0.0	0.0	0.2	0.4
	weak	<i>gauss2mf</i>	(0.0325 0.300 0.0325 0.400)	0.0325	0.2	0.3	0.4	0.5
	moderate	<i>gauss2mf</i>	(0.0325 0.500 0.0325 0.600)	0.0325	0.4	0.5	0.6	0.7
	strong	<i>gauss2mf</i>	(0.0325 0.700 0.0325 0.800)	0.0325	0.6	0.7	0.8	0.9
	over	<i>gauss2mf</i>	(0.0650 0.900 1.0000 1.000)	0.065	0.7	0.9	1.0	1.0
FJ	weak	<i>gauss2mf</i>	(1.000 0.000 0.065 0.300)	0.065	0.0	0.0	0.3	0.5
	moderate	<i>gauss2mf</i>	(0.065 0.500 0.065 0.700)	0.065	0.3	0.5	0.7	0.9
	strong	<i>gauss2mf</i>	(0.065 0.900 1.000 1.000)	0.065	0.7	0.9	1.0	1.0
FD	weak	<i>gauss2mf</i>	(1.000 0.000 0.065 0.200)	0.065	0.0	0.0	0.2	0.4
	moderate	<i>gauss2mf</i>	(0.065 0.400 0.065 0.600)	0.065	0.2	0.4	0.6	0.8
	strong	<i>gauss2mf</i>	(0.065 0.900 1.000 1.000)	0.065	0.6	0.8	1.0	1.0
FE	non	<i>gauss2mf</i>	(1.000 0.000 0.065 0.100)	0.065	0.0	0.0	0.1	0.3
	weak	<i>gauss2mf</i>	(0.065 0.300 0.065 0.400)	0.065	0.1	0.3	0.4	0.6
	moderate	<i>gauss2mf</i>	(0.065 0.600 0.065 0.700)	0.065	0.4	0.6	0.7	0.9
	strong	<i>gauss2mf</i>	(0.065 0.900 0.065 1.000)	0.065	0.7	0.9	1.0	1.0

As can be seen from Table 1, three values were assigned to σ in order to build any MF of our FSs, *i.e.* 1.000; 0.065 and 0.0325. Value of 1.000 is used only for MFs at the beginning and at the ending of any FS, and it has no influence in building MFs because its area of interest is outside the domain of MFs (e.g. the first parameter of FE's non or the third parameter of FE's strong for function *gauss2mf*; the second and fourth parameters of *gauss2mf* are lb and ub, respectively). Thus, the extremities of FSs (*i.e.* the beginning and ending of their domains) have σ value of 1.000, but it could have been any other value, since it does not affect the construction of any FS. This is the reason why we do not take it into account when presenting the figures in column σ of Table 1. This column displays only the value used to build each of the FSs, and it can be noticed that any MF has only one value for σ . Normally, function *gauss2mf* accepts two values for σ (first and third parameter), but we build symmetric MFs and we use only one value. The predominant value for σ in Table 1 is 0.065, which is a given value. As far as this value is concerned, and the reasons why it has been chosen, no details are available herein, since such a discussion would exceed the purpose of this article. The only reference made in this respect is that for more strict MFs (e.g. FC's moderate MF, on shorter domain than most of the other MFs) we use half of this predominant value of σ (*i.e.* 0.0325).

Figure 2 displays the FSs of our E-Democracy model, using the figures presented in Table 1. It may be observed that a given 0-1 scale and two values (*i.e.* 0.0325 and 0.065) for σ determine NME. By defining the rules of E-Democracy model (REDs) we can better understand the choice of building FSs when using the approach displayed in Figure 2. The eight REDs are :

- RED 1) **If (Citizenry is *non*) or (Justice is *weak*) then (E-Democracy is *non*)**
- RED 2) **If (Citizenry is *weak*) and (Justice is *not weak*) then (E-Democracy is *weak*)**
- RED 3) **If (Citizenry is *not non*) and (Justice is *moderate*) then (E-Democracy is *weak*)**
- RED 4) **If (Citizenry is *not non*) and (Justice is *not weak*) and (Delegates is *not moderate*) then (E-Democracy is *weak*)**
- RED 5) **If (Citizenry is *moderate*) and (Justice is *strong*) and (Delegates is *moderate*) then (E-Democracy is *moderate*)**
- RED 6) **If (Citizenry is *strong*) and (Justice is *strong*) and (Delegates is *moderate*) then (E-Democracy is *strong*)**

RED 7) If (Citizenry is over) and (Justice is not weak) and (Delegates is moderate) then (E-Democracy is weak)

RED 8) If (Citizenry is over) and (Justice is strong) and (Delegates is strong) then (E-Democracy is moderate)

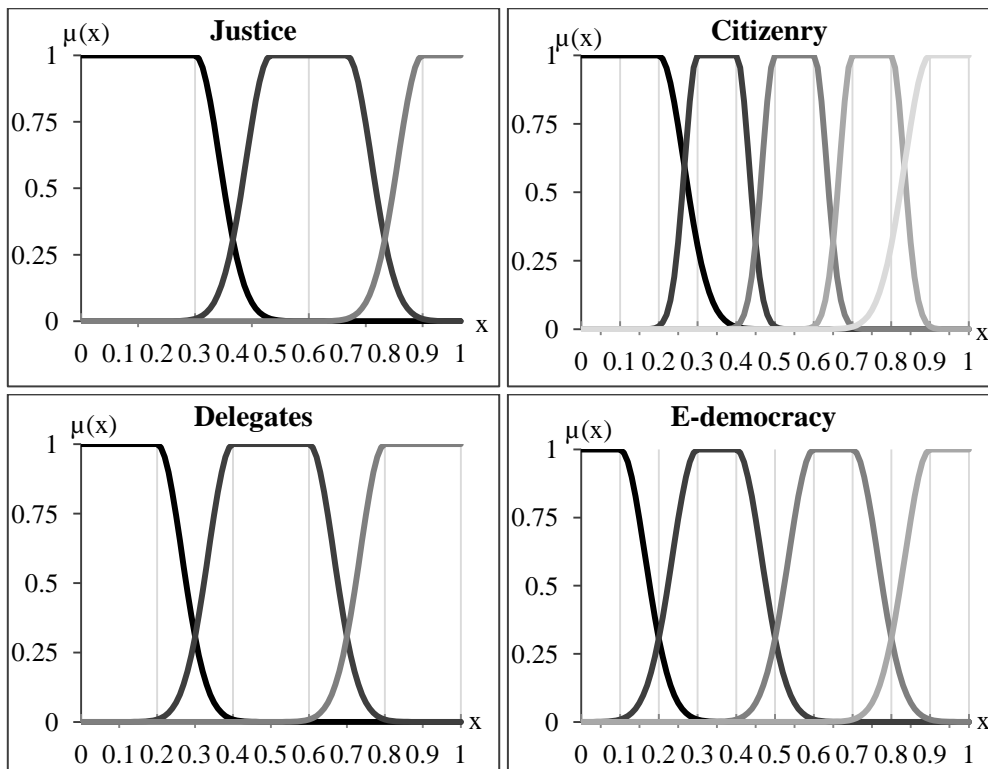


Fig 2: Fuzzy sets and fuzzy subsets of NME

Briefly speaking, through REDs, a model of E-democracy is proposed. In this model, Justice represents the watcher of all the other institutions, but Citizenry is decisive for obtaining high values of E-democracy output. In order to yield the theoretical maximum output, Delegates must be at a moderate level, Justice at high level and Citizenry must not overpass the limit of strong participation. A minimum output is theoretically obtained when either Justice or Citizenry are at their lowest level. A more elaborated description of E-democracy model has already been presented [6], and we do not want to insist in any other way on this subject.

Table 2 presents the mathematical formalization of REDs, using a Matlab approach [15]:

Table 2. Mathematical formalization of REDs. AND: 1; OR: 2

Antecedents			Consequents	Connectors
Citizenry	Justice	Delegates	E-democracy	AND/OR
1	1	0	1	2
2	-1	0	2	1
-1	2	0	2	1
-1	-1	-2	2	1
3	3	2	3	1
4	3	2	4	1
5	-1	2	2	1
5	3	3	3	1

In this section, we have only given a hint on what an MFIS is, and we have proposed a model of E-Democracy (*i.e.* NME), without discussing it in depth. NME will constitute an example when AOE is presented in the next section.

3. ALGORITHM FOR OPTIMUM E-DEMOCRACY (AOE)

This section is dedicated to an ample discussion about AOE, exploring through examples the MFIS of NME presented in section 2. The introduction of NME is made in order to prove that sometimes an objective function f cannot be available in a classic style. However, we try to demonstrate that a solution to optimize an MFIS is accessible, and this may help in future researches when MFIS is used, without needing to choose another substitute approach generated by the optimization problem. When the question of finding minimum and maximum outputs of E-Democracy arose in our previous research [6], the solution came naturally by using a human reasoning approach: successive trials which follow the steepest descent pattern. The steepest descent approach already proved to be convergent [4], but a mathematical proof of AOE convergence is not going to be presented in this paper. When AOE was conceived and implemented, methods of nonlinear differentiable optimization, especially backpropagation, were considered to be only an aspiration. Even if AOE could easily fall in the category of patterns of direct search methods (see section 1), the improvement (*i.e.* stepping up the search) was inspired by backpropagation alike methods which use gradients of first and second order [17] [18]. Only that AOE does not use gradients of f or of the least-square distance function, but explores a fluctuating region of each unknown value of vector x , and searches for a better value of f , thus determining intermediary solutions of f and x .

Figure 3 illustrates how AOE works in a basic approach; s is the fluctuant *step* used to create a region around any unknown value of vector x and ∇x is the vector combinations of all possible regions around any element of x .

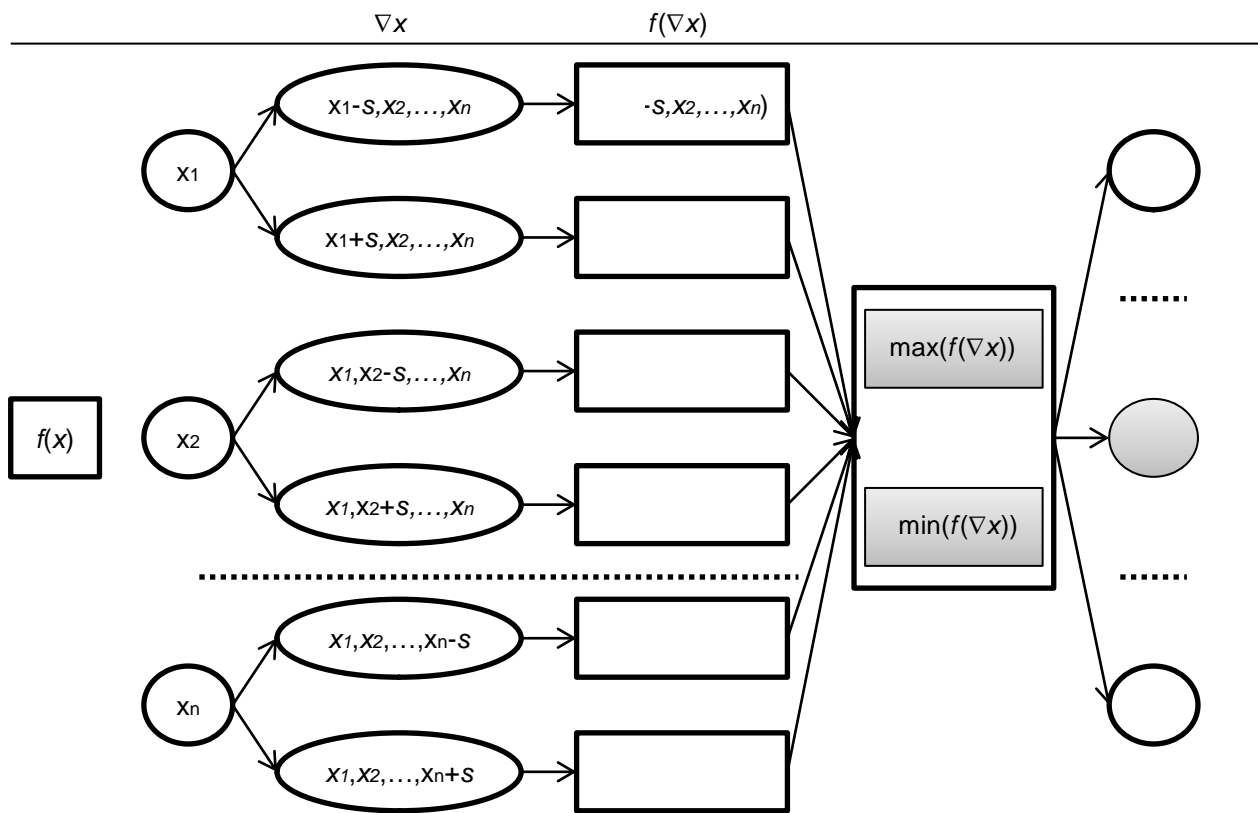


Fig 3: Basic AOE

The logic behind AOE is simple, the changed element of x (*i.e.* x' in Figure 3) that yields a better value for f takes the new value while the other elements of x remain unchanged. With each iteration we get new values for x and f and we stop the algorithm when some criterion is reached (e.g. maximum number of iterations or changed values of f less than a given precision). There is only one issue at this stage: how to adaptively use s or how to change it if no improvements of f are found after an iteration. Actually, this may be the key of any direct search method that follows the path of patterns. The solution is simple: we start from a given precision step, and we increase the searching step with each unsuccessful iteration. How to increase the step may raise another issue and there are several choices: add a given value to the step, double the step, using a logarithmic or exponential scale, using arithmetic, geometric or Fibonacci series etc. In our explorations the first two choices gave satisfying results, and we will discuss later this issue when presenting an improvement of AOE (IAOE).

A simple formalization of AOE has the following structure:

1. Let us read the type of optimization (*i.e.* minimization or maximization), the value of initial step s , the precision p , and the maximum number of iterations

2. Let us take the values of initial input vector x and calculate the initial value of output q
3. Let us initialize the final vector $x^* = x$, and let us initialize the final output $q^* = q$
4. Let us calculate, up to the limit of maximum number of iterations, new values of x^* and q^* using a fluctuant s to obtain a new ∇x on each iteration t . The condition of yielding new solutions of x^* and q^* is that $q_t^* + p < q_{t+1}^*$ for maximization and $q_t^* - p > q_{t+1}^*$ for minimization.

A more elaborated pseudo code formalization of AOE with an example applied on NME is also provided in Appendix A.1 and the Matlab implementation of AOE is available in Appendix A.2.

3.1. Supporting algorithms for AOE

Before presenting some results of AOE, let us have a short discussion on the Matlab formalization of REDs, illustrated in Table 2. In order to apply AOE we need to explore all the possible situations. Thus, the REDs containing negative inputs from Table 2 must be transformed in REDs with positive inputs (REDPIs). While from a linguistic and human reasoning point of view a formalization of REDs containing negative inputs is very convenient, from a computational point of view is burdensome. This is the reason why we need to transform all REDs in REDPIs and the result is displayed in Table 3. In order to achieve that, we use an algorithm that generates antecedents and consequents (AGAC), see Appendix B1 with examples.

Table 3. REDPIs, mathematically formalized, generated with AGAC. A-actual row, B-initial row, C-constraints, D-outputs, E-connectors; the highlighted lines are generated REDPIs, the non-highlighted lines are unchanged, mathematically formalized REDs

A	B	C			D	E
1	1	1	1	0	1	2
2	5	3	3	2	3	1
3	6	4	3	2	4	1
4	8	5	3	3	3	1
5	2	2	2	0	2	1
6	2	2	3	0	2	1
7	3	2	2	0	2	1
8	3	3	2	0	2	1
9	3	4	2	0	2	1
10	3	5	2	0	2	1
11	4	2	2	1	2	1
12	4	2	2	3	2	1
13	4	2	3	1	2	1
14	4	2	3	3	2	1
15	4	3	2	1	2	1
16	4	3	2	3	2	1
17	4	3	3	1	2	1
18	4	3	3	3	2	1
19	4	4	2	1	2	1
20	4	4	2	3	2	1
21	4	4	3	1	2	1
22	4	4	3	3	2	1
23	4	5	2	1	2	1
24	4	5	2	3	2	1
25	4	5	3	1	2	1
26	4	5	3	3	2	1
27	7	5	2	2	2	1
28	7	5	3	2	2	1

What AGAC has actually done in Table 3 is the elimination of operator 'not' so that all REDs are transformed in rules of E-democracy with positive inputs. One might easily observe that we have 28 REDPIs obtained from eight REDs. However, by combining three inputs in an exhaustive manner, we could only get 27 REDs. This is not a flaw of AGAC, but a redundancy given by the eighth RED, which is an exception to the seventh RED. It may be a natural thing in politics or social problems to deal with a lot of redundancies or discrepancies when specifying fuzzy rules of an MFIS. This also makes MFIS a choice for negotiating with uncertainties and human inaccurate perception in any society.

Another auxiliary tool for AOE is an algorithm that generates an initial solution (AIS), which is necessary for the second step of AOE. AIS finds the mean of each variable from any REDPI, or assign a neutral value for inputs that are unaccounted in a REDPI (*i.e.* has the value of zero). We are not going to insist on this, but we provide a formalization for AIS in Appendix B2.

3.2. Results of AOE

Up to this point, we have discussed AOE and we have introduced two other preliminary algorithms that support AOE (*i.e.* AGAC and AIS). It is now the moment of presenting some results obtained with AOE, taking advantage of the use of AGAC and AIS.

Table 4 presents results obtained with AOE for minimization and maximization of E-democracy output and also the number of iterations (NI) needed to reach the solution. These results prove that AOE finds E-democracy's minimum with a low precision $step = 0.01$, and further improvements cannot be found with a higher precision. Furthermore, the initial input solution x and initial output solution q are practically final solutions for inputs and output, respectively. On the contrary, because of REDs, we observe some improvements when maximizing the result of E-democracy. However, improvements

given by higher precision calculations are not important if we look from a political or social point of view, because final output q^* is identical for a precision of 10^{-4} no matter the value of $step$ from Table 4.

Figure 4 illustrates the effort needed by AOE to yield maximum output with different precision $step$, using a cubic interpolation of NI from Table 4.

Simulating a search for a best result on a computer cannot assure an extrapolation of AOE's results to a human universe. However, even if we think of it as a metaphor, we can take that a substantial easier human effort can determine if the output is a solid one, with a decent precision of a hundredth of the universe of discussion (*i.e.* $step = 0.01$ on a 0-1 scale). There is no need for an endeavor that would consume a lot of energy and would require exquisite skills to determine if E-democracy is far from its maximum. In case of minimization, we are not able to verify the accuracy of our E-democracy model, but we provide a comparison in section 5 with classical differentiable minimization methods in a series of data.

Table 4. Optimum results of E-democracy output using different precision $step$.
Maximization: $x=(0.75, 0.95, 0.50)$, $q=0.8566$; Minimization: $x=(0.10, 0.15, 0.50)$, $q=0.0931175$

$step$	Maximization					Minimization				
	x^*			q^*	NI	x^*			q^*	NI
0.01	0.7	0.99	0.4	0.897000	38	0.10	0.15	0.50	0.0931175	99
0.005	0.7	0.995	0.4	0.897010	78	0.10	0.15	0.50	0.0931175	199
0.001	0.699	0.999	0.4	0.897024	399	0.10	0.15	0.50	0.0931175	999
0.0005	0.6995	1	0.4	0.897027	800	0.10	0.15	0.50	0.0931175	2000
0.0001	0.6994	1	0.4	0.897028	4006	0.10	0.15	0.50	0.0931175	10000

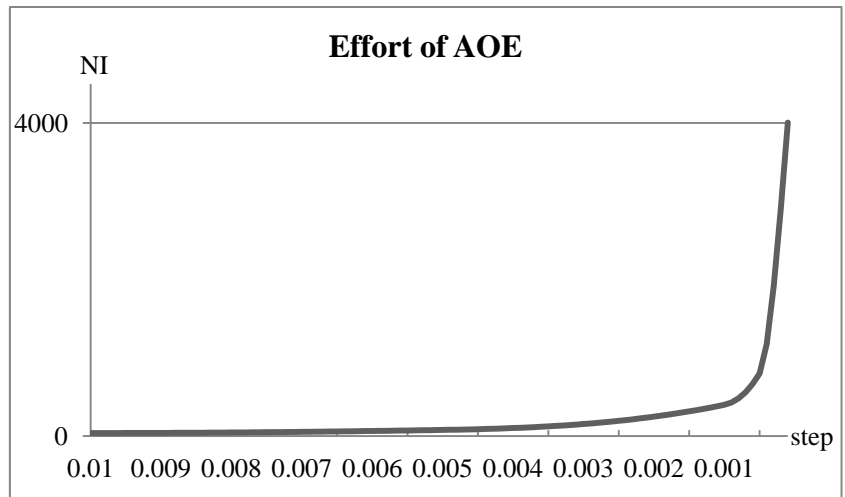


Fig 4: Exponential dependency between NI and $step$

Before discussing the results of E-democracy, we need to underline the fact that input solutions provided in Table 4 are only part of an infinity of solutions. Values of output determined with AOE are fixed, but values of x^* may vary. Thus if we fix two of the three inputs of E-democracy model, we can find the lower bound (LBFI) and upper bound (UBFI) of each of the final input, thus being able to determine the input intervals which yield the same optimized output q^* .

Table 5. LBFI and UBFI for a different precision $step$. C: Citizenry; J: Justice; D: Delegates

Input	$step = 0.01$			$step = 0.001$			$step = 0.0001$		
	LBFI	x^*	UBFI	LBFI	x^*	UBFI	LBFI	x^*	UBFI
C	0.6844	0.7	0.7	0.6931	0.699	0.7014	0.6973	0.6994	0.6997
J	0.8923	0.99	1	0.9365	0.999	1	0.97211	1	1
D	0.3906	0.4	0.525	0.3969	0.4	0.525	0.3989	0.4	0.5
q^*	0.897000			0.897024			0.897028		

We briefly present the algorithm that finds boundaries for each input (AIB) when all the other inputs are fixed. AIB is a bisection method extension, and a Matlab implementation of AIB is provided in Appendix B3:

- i. finding an initial lower bound (l) and initial upper bound (u) for x_i^* , $i = 1, n$
- ii. applying bisection method for both intervals: (l, x_i) and (x_i, u)
- iii. stop when $f(l)$ and $f(u) < f(x_i^*)$ for maximization and $f(l)$ and $f(u) > f(x_i^*)$ for minimization

When minimizing the E-democracy output, LBF1 equals zero and UBF1 equals one for all of the three inputs, no matter the value of *step*. This happens because of RED 1. When maximizing the output, we find non-trivial bounds for the inputs, as shown in Table 5.

As far as the results from Table 4 and Table 5 are concerned, we can now extract some conclusions. Firstly, identical maximum output for a precision of 10^{-4} (i.e. $q^* = 0.8970$) is yielded by large combinations of inputs, but all are in well-defined ranges, quite similar for different values of *step*. Secondly, it is obvious that for a higher *step* (i.e. lower precision) the range of all inputs is larger than for a smaller *step*, especially on the account of LBF1.

On the other hand, from a political point of view, E-democracy optimum output and inputs have significant similarities, no matter the value of *step*. Justice always tends to its maximum possible value (value of 1 on a 0-1 scale). For a decent precision from a human perspective, *step* = 0.01 (i.e. 1% on a 0-1 scale), lower values of Justice, down to 0.8923, yield an optimum output $q^* = 0.8970$ when Citizenry and Delegates are fixed, i.e. $x^* = (0.7, 0.8923, 0.4)$. Thus, the contribution of Justice to an optimum E-democracy is always in the range of its crisp values of *strong* Justice. Reminding that RED 6 clearly states *strong* Citizenry, *strong* Justice and *moderate* Delegates for a *strong* E-democracy, we also observe that Delegates has optimum values in a range that is similar to its first half of *moderate* support (see Table 1 and Figure 2). On the contrary, Citizenry must be on a range that assure its *strong* level, but avoids the *over* level (see Table 1 and Figure 2). This means that it always takes fuzzy values of *strong* Citizenry in order to obtain a maximum E-democracy.

In conclusion, Justice must always be at a *strong* level, Delegates and Citizenry must reduce their *moderate* and *strong* levels, respectively, so that E-democracy could reach its maximum point. On the other hand, minimum E-democracy is easier to be achieved, if either Citizenry or Justice is at its lowest level. Nevertheless, none of the minimum and maximum outputs reaches the bounds of E-democracy's 0-1 scale in our model. We also observe that the value of 0.8970 for q^* empirically proves that E-democracy, as democracy in general, is not a perfect system, but perfectible. What variables of E-democracy model influence the optimum output will be part of a future research.

3.3. Improvement of AOE (IAOE)

While AOE uses a constant value of step *s*, IAOE tries to speed up the search by adaptively changing *s*. The logic behind IAOE is based on finding a value of an input where no improvements of the objective function *f* can be obtained. Instead of searching with small but solid steps for a solution, we try to find a better solution by decreasing and then incrementally modifying the step *s*. Figure 5 illustrates the basic of IAOE, and we may recognize the pattern of bisection method when searching for an improved *f*.

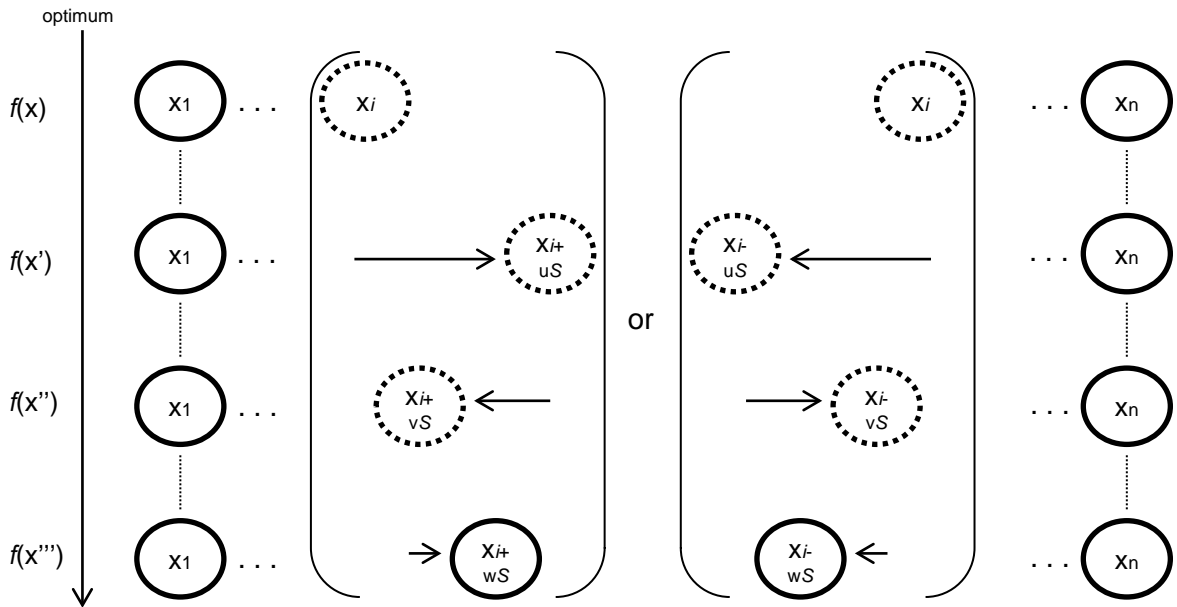


Fig 5: Basic IAOE. $i \in \{1, 2, \dots, n\}$

The pseudo code of IAOE is provided in Appendix C1 and the Matlab implementation is also provided in Appendix C2. A more simple formalization of IAOE is described in the following steps:

- i. taking step *S*, and reading the initial values of *x* and *q*
- ii. initializing the final input $x^* = x$
- iii. calculating the bound, i.e. new input x' , by sequentially increasing the value of step *S*
- iv. applying bisection method for the interval given by initial value (i.e. x_i) and bound (i.e. $x_i + uS$ or $x_i - uS$)

Table 6 presents optimum inputs and outputs obtained with AOE and IAOE using the same variables as in Table 4, which only shows results yielded by AOE alone. The number of iterations NI presented in Table 6 is a total number given by AOE and IAOE. Furthermore, IAOE has two parts that require multiple iterations: finding x^* (i.e. IAOE 3) and finding x (i.e. IAOE 4). If we compare results from Table 5 with results from Table 6, we observe two important things. Firstly, minimization is achieved with the same optimum output in infinitesimal steps when using IAOE. Secondly, for maximization, we have fewer steps with IAOE but the optimum output is not as precise. The difference appears when maximizing, because of the precision $step$, which is the same in both AOE and IAOE.

Table 6. Optimum results of E-democracy output with AOE and IAOE.
Maximization: $x=(0.75, 0.95, 0.50)$, $q=0.8566$; Minimization: $x=(0.10, 0.15, 0.50)$, $q=0.0931175$

step	Maximization					Minimization				
	x^*			q	NI	x^*			q	NI
0.01	0.6825	0.950	0.5	0.896567	15	0.10	0.15	0.50	0.0931175	7
0.005	0.6912	0.950	0.5	0.896567	29	0.10	0.15	0.50	0.0931175	8
0.001	0.6975	0.937	0.5	0.896935	83	0.10	0.15	0.50	0.0931175	10
0.0005	0.6975	0.947	0.5	0.896909	72	0.10	0.15	0.50	0.0931175	11
0.0001	0.6996	0.972	0.5	0.897016	130	0.10	0.15	0.50	0.0931175	14

Table 7 presents a comparison between AOE and IAOE, using the same value for $step$, and a comparison between AOE and IAOE, using a smaller precision for IAOE.

Table 7. Comparison of maximization with different precisions for AOE and IAOE

step _{AOE}	Maximization: step _{IAOE} = step _{AOE}					Maximization: step _{IAOE} = step _{AOE} × 0.01				
	x^*			q	NI	x^*			q	NI
0.01	0.6825	0.950	0.5	0.896567	15	0.7002	0.9711	0.5	0.896995	87
0.005	0.6912	0.950	0.5	0.896567	29	0.7002	0.9805	0.5	0.897002	120
0.001	0.6975	0.937	0.5	0.896935	83	0.6991	0.9927	0.5	0.897017	135
0.0005	0.6975	0.947	0.5	0.896909	72	0.6995	0.9961	0.4003	0.897029	285
0.0001	0.6996	0.972	0.5	0.897016	130	0.6994	0.9992	0.4	0.897028	414

Results from Table 7 prove that we may trade NI for a better precision if we use a value of $step$ for IAOE which is a hundredth of the value of AOE's $step$. These results are also comparable with results from Table 4, where AOE yields optimum inputs and output without IAOE, but with far more NI.

In order to prove IAOE consistency, we need to compare LBFI and UBFI of AOE with IAOE and AOE alone, for different precision $step$. Table 8 shows the range of each input when the other two are fixed and the output is maximum, using AIB.

Table 8. Maximization using AOE with IAOE: comparison of results using different precision $step$.
A: step_{IAOE} = step_{AOE}; B: step_{IAOE} = step_{AOE} × 0.01; C: Citizenry; J: Justice; D: Delegates

A	step = 0.01			step = 0.001			step = 0.0001		
Input	LBFI	x^*	UBFI	LBFI	x^*	UBFI	LBFI	x^*	UBFI
C	0.6825	0.6825	0.7225	0.6975	0.6975	0.6975	0.6991	0.6997	0.6997
J	0.93	0.99	1	0.9365	0.9365	1	0.9719	1	1
D	0.3975	0.4	0.55	0.3997	0.5	0.5	0.3997	0.5	0.5
q^*	0.896567			0.896938			0.897016		

B	step = 0.01			step = 0.001			step = 0.0001		
Input	LBFI	x^*	UBFI	LBFI	x^*	UBFI	LBFI	x^*	UBFI
C	0.6827	0.7	0.720	0.6931	0.699	0.7003	0.6973	0.6994	0.6997
J	0.8911	0.97	1	0.9364	0.993	1	0.9721	0.9992	1

D	0.39	0.5	0.58	0.3969	0.5	0.54	0.3989	0.4	0.5
q	0.896995			0.8970166			0.897028		

By comparing Table 8 and Table 5, we observe that LBFIs and UBFIs tend to be similar when optimum inputs and output are similar. AOE is a heuristic algorithm that yields decent results which depend on the precision *step*. AIB and IAOE also depend on AOE, because it offers the solution (*i.e.* optimum and intermediary inputs and output) based on which they search for bounds or improvements. When using IAOE to accelerate the search for optimum results, we are closer to the results yielded by AOE alone only if we use $step_{IAOE} = step_{AOE} \times 0.01$. There are also some inconsistencies of AOE with IAOE and two of them can be spotted in Table 7. Firstly, when $step_{IAOE} = step_{AOE}$, for a lower precision $step = 0.001$ we obtain a better result (but with more NI) than with $step = 0.0005$. Secondly, when $step_{IAOE} = step_{AOE} \times 0.01$, we reach a better output with $step = 0.0005$ than with $step = 0.0001$ and with less NI. These inconsistencies are normal in a heuristic approach which always depends on initial solutions, on precision and, to a greater degree, on the type of problem it deals with.

From a human perspective point of view and with decent precision, we believe that AOE and all the other components may lead to consistent results in an MFIS, especially when dealing with problems related to common knowledge. In the next section, we will try to verify AOE in a context that is not quite appropriate for it. We do this in order to examine the behavior of AOE and its components in harder environmental conditions, where differentiability of least-square function is appropriate.

We conclude this section by wrapping up its contents: we have described a direct search algorithm which offers a reliable solution to an MFIS (*i.e.* AOE) and we have also discussed some other algorithms that support (*i.e.* AGAC, AIS), improve (*i.e.* IAOE) and broaden (*i.e.* AIB) the AOE introduced herein. We have applied AOE and its family of methods to the MFIS of NME, and we have obtained some interesting results which are going to be summarized in Conclusions.

4. COMPARING AOE WITH DIFFERENTIABLE MINIMIZATION METHODS

In this section, we will compare AOE to a method of nonlinear differentiable minimization for a series of data. The classical approach of nonlinear optimization takes into account the minimization of Euclidian distance of a given series of data against a theoretical analytical function, see formula (2).

$$d(C,F(x)) = 2 \sqrt{\sum_{i=1}^n (C_i - F_i(x))^2}, \quad d(C,F(x)) \text{ is minimum} \quad (2)$$

$d(\cdot, \cdot)$ is Euclidian distance, C is a vector of n observations

F is a theoretical function and x is a vector of parameters

We will use the Matlab function *lsqnonlin* which implements two algorithms: Levenberg-Marquardt (LM) [17] and Newton interior reflective trusted-region (TR) [18]. We refer to previous research in the field of option pricing estimation through one parameter and three parameters [19]. We will try to verify if AOE is consistent with non-linear minimization which yields an implied volatility or implied volatility, skewness and kurtosis. In case of implied volatility, the function that models one vector of observed variables from a series of data is given by Black-Scholes formula [20]. In case of the three implied parameters: volatility, skewness and kurtosis, the function that models a vector of observed variables in a series of data is given by Negrea-Maillet-Jurczenko formula [21]. Appendix D describes in detail the methods, data and results used in this short comparison between AOE, LM and TR for a minimization process. Table 9 presents transactions of two series of call options data (*i.e.* O1 and O2).

Table 9. Transactions for two options: O1 and O2.
C: observed call option price; K: strike price; S: underlying price; r: interest rate; T-t: duration

O1					O2				
C	K	S	r	T-t	C	K	S	r	T-t
510	2500	2979.46	3.4	75	195	2500	2608.54	3.32	217
440	2500	2904.27	3.41	74	195	2500	2610.05	3.32	217
437	2500	2898.15	3.41	74	170	2500	2562.56	3.38	194
433	2500	2892.12	3.41	74	172	2500	2567.17	3.38	194
430	2500	2889.95	3.41	74	174	2500	2574.46	3.38	194
380	2500	2837.48	3.41	71	174	2500	2573	3.38	194
380	2500	2838.53	3.41	71					
524	2500	2987.6	3.38	69					

525	2500	3004.29	3.38	69
531	2500	3006.65	3.38	69

Using formula (2), we will minimize the data of O1 and O2 for one and three parameters, see also Appendix D. Table 10 presents the results obtained for one parameter (*i.e.* volatility) minimization:

Table 10. Minimization for one parameter

Method	O1: initial solution $x = 0.5$					
	x^*	d	step	NI	absolute/relative error	
AOE	0.2928348	130.17481188	0.00001	20001	-8.31E-06	-0.000006%
+	0.292856	130.17494886	0.0001	10552	-1.45E-04	-0.000112%
IAOE	0.29258	130.20802619	0.001	1054	-3.32E-02	-0.025515%
LM	0.292838912	130.17480358	-	-	0.00E+00	0.000000%
TR	0.292838892	130.17480358	-	-	0.00E+00	0.000000%
AOE	0.29284	130.17480419	0.00001	20001	-6.10E-07	0.000000%
	0.2928	130.17555411	0.0001	12072	-7.51E-04	-0.000577%
	0.293	130.18769589	0.001	1206	-1.29E-02	-0.009903%

Method	O2: initial solution $x = 0.5$					
	x^*	d	step	NI	absolute/relative error	
AOE	0.1380796	279.40903389	0.00001	20001	-7.50E-07	0.000000%
+	0.1380505	279.41132961	0.0001	11459	-2.30E-03	-0.000822%
IAOE	0.1376179	279.96551909	0.001	1035	-5.56E-01	-0.198769%
LM	0.1380801	279.40903314	-	-	0.00E+00	0.000000%
TR	0.1380801	279.40903314	-	-	0.00E+00	0.000000%
AOE	0.13808	279.40903319	0.00001	20001	-5.00E-08	0.000000%
	0.1381	279.41006166	0.0001	13619	-1.03E-03	-0.000368%
	0.138	279.42576994	0.001	1361	-1.67E-02	-0.005990%

Both O1 and O2 offer decent results for the final solution x^* and for the Euclidian distance d . We observe that, with a single exception, AOE alone yields better results than AOE with IAOE (using $step_{IAOE} = step_{AOE} \times 0.01$). In case of minimization through Euclidian distance the difference between NI is not as large as in case of minimization of MFIS, see Table 6. When using $step_{IAOE} = step_{AOE} \times 0.1$ and $step_{IAOE} = step_{AOE}$ we also obtain surprisingly good results, see Appendix D.

So far, for one parameter minimization, AOE yields decent results, although with a large amount of time, as given by NI, see Table 10. The next step is to see if AOE may lead to decent results for a three parameters minimization. In case of one parameter minimization the results are in the range of precision $step$. AOE cannot use a differentiability of any function, either the Euclidian distance or any other function that models the data (when using an algorithm like backpropagation). AOE is designed for MFIS, but it may be applied to some other types of fuzzy inference system (*e.g.* Sugeno-Takagi type), and it works well for minimization and maximization. This is the reason why results for multiple minimization are far better achieved with algorithms designed to work with Euclidian distance. Even with such methods (*i.e.* LM and TR), we have no guarantee for optimum results.

Table 11 presents a comparison between AOE, LM and TR for multiple (volatility, skewness and kurtosis) non-linear minimizations of O1 and O2. Results from Table 11 give an indication of what AOE can do for multiple parameters minimization. Sometimes, even for LM (in case of O1) or TR (in case of O2) is difficult to yield a decent solution. Although values of final vector solution x^* are not the same for AOE with IAOE comparing to TR or LM, they sometimes yield a good and very good minimized Euclidian distance d .

The goal of comparing AOE with TR and LM is to prove that, through a heuristic approach, it is consistent with its results. For minimization AOE depends on the initial solution, but so does LM and so does TR when it comes to bounds of the final solution. AOE is not designed for minimization using Euclidian distance, but it has the ability of finding decent solutions, especially for one parameter minimization. Without an alternative for minimization or maximization of an output of MFIS,

we have only verified AOE in the context of a problem for which AOE is not designed. However, this also proves some sort of consistency and accuracy of AOE.

In this section we have compared AOE and AOE with IAOE to non-linear optimization least-square methods and we have observed that for one parameter minimization the results of AOE (with or without IAOE) are very good, while for three parameters minimization they occasionally yield decent solutions.

Table 11. Three parameters non-linear minimization

Method	O1: initial solution $x = (0.5; -1; 1)$							
	\bar{x}			d	step	NI	absolute/relative error	
AOE	0.17	-8.29	6.63	122.9855	10^{-5}	20001	-2.01E+00	-1.6332%
+	0.18	-7.64	1.37	123.5460	10^{-4}	11768	-2.57E+00	-2.0795%
IAOE	0.18	-7.85	2.77	123.3385	10^{-3}	1227	-2.36E+00	-1.9147%
LM	NaN	NaN	NaN	NaN	-	-	NaN	NaN
TR	0.98	-1.34	-9.71	120.9769	-	-	0.00E+00	0.0000%
AOE	0.14	-2.52	1.00	3611.7008	10^{-5}	20001	-3.49E+03	-96.6504%
	0.14	-2.52	1.00	3611.7008	10^{-4}	20001	-3.49E+03	-96.6504%
	0.19	-7.35	-0.29	123.8450	10^{-3}	9230	-2.87E+00	-2.3159%

Method	O2: initial solution $x = (0.5; -20; 1)$							
	\bar{x}			d	step	NI	absolute/relative error	
AOE	0.04	-27.60	-2.32	0.4821	10^{-5}	20001	-1.90E-01	-39.5072%
+	0.04	-25.97	-5.77	0.3539	10^{-4}	11040	-6.22E-02	-17.5796%
IAOE	0.04	-15.18	-23.63	0.5588	10^{-3}	1680	-2.67E-01	-47.8062%
LM	0.04	-22.54	-11.93	0.2917	-	-	0.00E+00	0.0000%
TR	0.19	15.38	-13.99	7.9592	-	-	-7.67E+00	-96.3352%
AOE	0.50	-20.00	0.81	12837.0470	10^{-5}	20001	-1.28E+04	-99.9977%
	0.45	-20.00	-0.95	10270.9820	10^{-4}	20001	-1.03E+04	-99.9972%
	0.05	-21.37	-3.72	77.4435	10^{-3}	20001	-7.72E+01	-99.6234%

CONCLUSIONS

This paper has presented a direct search algorithm (*i.e.* AOE) which finds the optimum output in a fuzzy inference system (*e.g.* MFIS). We have introduced a model of E-Democracy (*i.e.* NME) that is based on an MFIS and we have applied AOE to find its optimum outputs. We have also provided other algorithms for: generating the positive antecedents and constraints of an MFIS (*i.e.* AGAC), finding an initial solution for AOE (*i.e.* AIS), searching for boundaries of solutions (*i.e.* AIB), and accelerating the search for solutions (IAOE). We have compared AOE to differential least-square methods of optimization and we have observed that AOE may yield decent solutions on some occasions, especially for one parameter optimization.

AOE depends on some variables: initial solution, precision and number of iterations. Results are not strongly affected by these variables in case of fuzzy optimization, while for least-square minimization the variables may be decisive. Nevertheless, our field of interest is non-differentiable optimization, and AOE and its supportive components find a global optimum solution. This optimum solution may occur in an unpredictable range, and we have proved that in an MFIS there may be an infinity of solutions on a certain interval (using AIB). When speeding up the search (using IAOE) the number of iterations substantially decrease, but on some occasions we may obtain a poorer result than searching with smaller steps. The precision is important for both accelerated and simple search, but the improvements of the objective function lie in the range of that precision, which makes AOE (with or without IAOE) reliable.

Further researches on direct search methods for fuzzy optimization may try to better adapt the step that modifies the values of intermediary solution vector. We identify two approaches for improvements: statistics and predictions on objective function curvature, and random sampling of intermediary vector solutions.

REFERENCES

- [1] C. A. Floudas and P. M. Pardalos, *Encyclopedia of Optimization*, 2nd ed. New York: Springer, 2009.
- [2] T. G. Kolda, R. M. Lewis, and V. Torczon, "Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods," *SIAM Review*, vol. 45, no. 3, pp. 385–482, August 2003.
- [3] Robert Hooke and T.A. Jeeves, "'Direct Search' Solution of Numerical and Statistical Problems," *Journal of the Association for Computing Machinery*, vol. 8, no. 2, pp. 212-229, April 1961.
- [4] Robert Michael Lewis, Virginia Torczon, and Michael W. Trosset, "Direct search methods: then and now," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 191-207, 2000.
- [5] E.H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1-13, 1975.
- [6] Ovidiu Turcoane, "Defining E-Democracy's Strength Using Fuzzy Logic Rules," in *Proceedings of the Eleventh International Conference on Informatics in Economy Education, Research & Business Technologies*, Bucharest, 2012, pp. 544-548.
- [7] Robert Fullér, *Fuzzy Reasoning and Fuzzy Optimization*. Turku, Finland: Akademi University, 1998.
- [8] Andreas Maier, *eDemocracy & eGovernment. Stages of a Democratic Knowledge Society*. Berlin, Germany: Springer, 2012.
- [9] UNESCO, *Towards Knowledge Societies*. Paris, France: UNESCO Publishing, 2005.
- [10] Robert Axelrod, *The Evolution Of Cooperation*. New York: Basic Books, 1984.
- [11] Jean-Jacques Rousseau, *The Social Contract Or Principles Of Political Right*. Whitefish, USA: Kessinger Publishing, 2004.
- [12] John Locke, *Second Treatise Of Government*. Indianapolis, USA: Hackett Publishing, 1980.
- [13] Charles-Louis de Secondat Baron de La Brède et de Montesquieu, *The Spirit of Laws*. Kitchener, Canada: Batoche Books, 2001.
- [14] John Rawls, *Political Liberalism*. New York, USA: Columbia University Press, 1996.
- [15] Matlab. (2014, Feb.) MathWorks. [Online]. <http://www.mathworks.com/help/fuzzy/index.html>
- [16] Lotfi Askar Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338-353, June 1965.
- [17] Donald W. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *SIAM Journal on Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [18] Thomas F. Coleman and Yuying Li, "On the convergence of interior-reflective Newton methods for nonlinear minimization subject to bounds," *Mathematical Programming*, vol. 67, no. 1-3, pp. 189-224, October 1994.
- [19] Ovidiu Turcoane, "Option Price Estimations and Speculative Trading In Knowledge Society," *Informatica Economica*, vol. 16, no. 4, pp. 131-141, 2012.
- [20] Fischer Black and Myron Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637-654, May-June 1973.
- [21] Bogdan Negrea, Bertrand Maillet, and Emmanuel Jurczenko, "Skewness and Kurtosis Implied by Option Prices: A Second Comment," Financial Markets Group. School of Economics and Political Science, London, Discussion paper 419, 2002.

Appendix

- A1. <http://turcoane.com/ijct/AOE-pseudocode.pdf>
- A2. http://turcoane.com/ijct/AOE_IAOE.txt
- B1. <http://turcoane.com/ijct/AGAC-pseudocode.pdf>
- B2. <http://turcoane.com/ijct/AIS-pseudocode.pdf>
- B3. <http://turcoane.com/ijct/AIB.txt>
- C1. <http://turcoane.com/ijct/IAOE-pseudocode.pdf>
- C2. <http://turcoane.com/ijct/IAOE.txt>
- D. <http://turcoane.com/ijct/Comparing-AOE-with-other-minimization-methods.pdf>