Engineering Letters, 13:2, EL_13_2_10 (Advance online publication: 4 August 2006)

The Human Evolutionary Model: A new approach for solving nonlinear optimization problems avoiding the problem of cycling

Oscar Montiel, Oscar Castillo, Jose Soria, Antonio Rodriguez, Hector Arias, Roberto Sepulveda

Abstract— **The aim of this paper is to give a method for reducing the problem of getting trapped in local optima (cycling), which is a common problem in evolutionary algorithms. For solving this problem we are proposing to use a Tabu method for avoiding already visited regions, this in combination with a novel fuzzy method that can handle imperfect knowledge in a broader way than Intuitionistic fuzzy logic does. This fuzzy method can manage non-contradictory, doubtful, and contradictory information provided by experts, providing a mediated solution, so we called it Mediative Fuzzy Logic.**

Index Terms—**Evolutionary Algorithms, Fuzzy Logic, Optimization.**

I. Introduction

A fundamental issue in any searching problem is the concept of a solution. This concept is very important, since we need it to indicate in the search space the locations where solutions are, if there are any. Without caring for the characteristics of the problem, the solution concept partitions the search space in two broad classes: solutions, and no-solutions. Generally, we can distinguish between these two classes by applying systematically a test for measuring properties that can be present or absent in the selected locations. Real world search problems can be established as optimization problems, since usually they are very difficult because they can have a large and multimodal search space with several solutions, it is very common to accept as good a solution which satisfies the problem rather than getting the optimal one. A search space can have an exponential number of solution concepts applied to it. We obtain a particular search problem when we applied a particular solution concept to a search space. Solution concepts are intrinsic to search problems, but we have to select and implement from a great variety of searching methods, an algorithm to obtain solutions for solving different search problems. For consistency, algorithms in the same search space must implement the same solution concept, although it is known that some algorithms may be more or less efficient than others. Algorithms that implement different solution concepts solve different search problems [1].

Many search problems require the optimization of a function $f{:}A\rightarrow\mathbf{R}$, i.e., we want to optimize a given real valued function f which is called the *objective function* or *cost function*, and to find a feasible solution that minimizes or maximizes this function is called and *optimal solution*. Typically A is some subset of the Euclidean space $\mathbf{R}^{\mathbf{n}}$, often specified by a set of constrains, equalities or inequalities that the members of A have to satisfy, in other words, when $A\in\mathbf{R}^{\mathbf{n}}$ we have a *constrained* optimization problem, and A is called the *constrained set* or *feasible set*. At the other hand, when $A=\mathbf{R}^{\mathbf{n}}$ we have an *unconstrained* optimization problem. Considering the next optimization problem:

$$\begin{array}{ll}\text{Minimize} & f(x)\\ \text{Subject to} & \boldsymbol{x}\in A\end{array}$$

where $\boldsymbol{x}$ is an n-vector of independent variables, $\boldsymbol{x}=[x^1,x^2,\ldots,x^n]^T\in A$, and the variables $x^1,x^2,\ldots,x^n$ are referred to as *decision variables*. This optimization problem can be viewed as a decision problem that involves finding the "best" vector $\boldsymbol{x}$ of the decision variable over all possible vectors in A. In this case the vector x is called the minimizer of f over A. There are also optimization problems that require maximization of the objective function [2].

There are several optimization methods that has been fully addressed in the literature [2,3,4,5]. In this paper we are using as optimization method the Human Evolutionary Model (*HEM*), which is a novel global optimization method presented for the first time in [6].

In any evolutionary algorithm we have many parameters to adjust, and generally they are adjusted by trial and error. There are many parameters used in evolutionary algorithms, for example, initial population size, number of crossover points,

Manuscript received December 9, 2005 and accepted April 5, 2006. This work was supported in part by the Research Council of DGEST under Grant 493.05-P. Students also were supported by CONACYT with scholarships for their graduate studies.

Oscar Montiel is with CITEDI-IPN Tijuana, Mexico (e-mail: oross@citedi.mx), and he is a Ph. D. student of UABC University.

Oscar Castillo is with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico (corresponding author phone: 52664-623-6318; fax: 52664-623-6318; e-mail: ocastillo@tectijuana.mx).

Patricia Melin is with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico (e-mail: pmelin@tectijuana.mx)

Antonio Rodriguez is with the Division of Graduate Studies and Research of UABC University, Mexico (e-mail: ardiaz@uabc.mx).

Hector Arias is with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico (e-mail: harias@tectijuana.mx).

Roberto Sepulveda is with CITEDI-IPN Tijuana, Mexico (e-mail: rsepulve@citedi.mx) and he is a Ph. D. student of UABC University.

and the probabilities for the mutation or crossover operators. Typically, each parameter is adjusted one at time, since often it is unknown how they interact, which may lead to sub-optimal choices, since often it is not known how the parameters interacts. Parallel tuning of multiple parameters can cause a big increment in the size of test that we have to perform. One alternative solution is to transfer the resulting parameters of a given problem to a new similar one, but there is no guarantee to obtain good result using method. The rigid form of static parameters contradicts the dynamic nature of evolutionary algorithms (EAs), for example, it will be desirable to use a large mutation step in early generations to get a faster approximation, and change values to obtain a more accurate solution when we detect that we are near to it. Unfortunately, this is not possible with fixed parameters, so an intuitive approach is to evolve the parameters with the algorithm at the same time. Many different procedures have been researched to adapt the parameters. It is common that the parameter choice differs strongly from case to case, but the main idea is to no longer choose the parameters semi-arbitrarily but to let the parameters to auto adapt. Self-adaptation is a phenomenon, which makes evolutionary algorithms flexible and closer to natural evolution [1].

HEM is a "Self-adaptive algorithm that evaluates its own behavior and changes its behavior when the evaluation indicates that it is not accomplishing what the algorithm is intended to do, or when better functionality or performance is possible"[7]. HEM has the skill of avoiding getting trapped in the same region of the landscape, as well to promote the evolution towards optimal solution.

The method for reducing the cycling problem has been developed for HEM, so we are giving an explanation of this novel evolutionary method in section II. In section III, we are explaining the Mediative Fuzzy Model and its relation with traditional and Intuitionistic Fuzzy Models. Section IV is devoted to show some experimental results focusing in the inference system, and finally in Section V we are giving conclusions.

II. The Human Evolutionary Model

HEM has been defined as an eight tuple

$$HEM = \left(AIIS, P, O, S, E, L, TL, VRL\right) \tag{1}$$

where

AIIS	Adaptive Intelligent Intuitive System
P	*P*opulation of human *N* like individuals
O	Single or a multiple objective goals,
S	Evolutionary strategy used for reaching the objectives expressed in *O*
E	Environmental, here we can have predators, etc.
L	Landscape, i.e., the scenario where the evolution must be performed
TL	Tabu List formed by the best solution found,
VRL	Visited Region List

In Fig. 1 we are showing a schematic representation of one individual in *HEM*. It has three parts: a genetic representation *gr*, which can be codified using binary or floating-point representation; a set of genetic effects *ge*, that are attributes of each individual such as "physical structure", "gender", "actual age", "maximum age allowed", pheromone level", etc; these attributes give to the algorithm some of the human like characteristics that will define in great part, the individual behavior. The third part in the individual representation is devoted to individual's fitness values. An individual p_i is defined as $p_i=(gr_i,ge_i,fv_i)$ where $gr_i=(gr_{i1},\ldots,gr_{iM})$ is a vector (a row) of the matrix *GR* of dimension $M \times N$. The genetic effects (ge_i) are rows in a matrix *GE*. In this method we can have one or several fitness values (*fv*), so we can handle single objective optimization problems (SOOP), and multi-objective optimization problems (MOOP). Fitness values are defined as vectors fv_i in the matrix $FV_{J\times N}$, in this way we have $fv=(fv_1,\ldots,fv_J)$. In this context, a population P_i is defined as $P_i=(GR_i+GE_i+FV_i)$. In the attribute $ge_{igender}$, we have the valid values set $\{M,F,0\}$, in this set *M* alludes a subpopulation of male individuals, *F* is used for the female subpopulation, and *0* means that this attribute will not be considered.

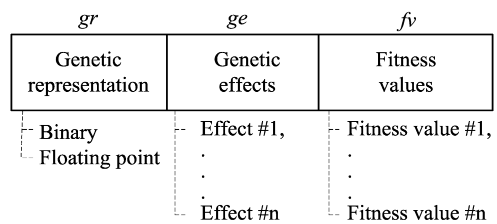

Fig. 1: Representing one individual in *HEM.*

The genetic attribute $ge_{iactAge}$ contains the actual age of an individual; its value corresponds to the number of generation that the individual has survived. We can set the maximum life expectance for each individual in the attribute $ge_{imaxAge}$.

The task of the attribute $ge_{iphLevel}$ is to leave trace about which individuals have been involved in previous generations producing good offsprings. Fig. 2 shows the general structure of *HEM*. The method consists in:

1. Create an initial population P_0 of size *N*. Here, we are going to create GR_0 and GE_0 of population P_0. The programmer must provide the range of each coefficient h_i in gr_i for creating appropriately GR_0. In the same way, the attributes of GE_0 will be set.
2. Evaluate GR_0. In this stage we are going to assign the corresponding fitness values (fv_i) to each gr_i. With this step, we completed the creation of P_0. Sort P_0 in ascending order using FV_i.
3. Repeat steps 3 to 20 until we fulfill a termination criterion.
4. Apply to the whole population P_i the "Genetic effect operator # 1". This operator works on GE_i, it will add "1" to "actual age" $(ge_{iactAge})$ of each individual of the population.
5. We apply the operator "Predator # 1" to P_i. This operator verifies the age of each individual of P_i, it will kill individuals that reach the attribute $(ge_{imaxAge})$.
6. The task of "Genetic effect operator # 2" is to mutate some of the genetic effects of individuals. Functionally, the most evident is to use $ge_{igender}$.

7. The operator "Predator # 2" will analyze the actual population to verify the gender balance; we want to know if the population of male and female individuals is balanced, or at least it is into a valid rate. If the population is balance "Predator #2" will do not carry out any action, but if the population is out of balance, this operator will proceed to balance it by predating the dominant subpopulation. For achieving this process, we have to select randomly as many individuals as we need and change its gender. We preferred to change the gender of individuals instead of killing them because in the process of eliminating individuals we could lose some good individuals.
8. At each generation, the best individual and its fitness values (values in MOOP) are saved into a list; this list is actually a Tabu list (*TL*) where previously visited solutions are stored.
9. Select individuals according pyramidal rule. *HEM* has a flexible selection process driven by an adaptive intelligent/intuitive system, which can manipulate any parameter involved in this process. Fig. 3 shows a distribution in quantity of individuals selected for creating a new population. The variable $S(g)$ represent the size of this subpopulation at generation g. In this figure, we are showing two ways for creating this new subpopulation, and it is controllable by *AIIS* using the state (enable/disable) of the variable *TS*. When *TS* is disable, we select as parents a percentage of the best individuals of the actual generation $s_1(g)$, plus the best individuals selected using a special polarized random distribution for favoring individuals with the highest pheromone level and fitness value $s_2(g)$, and the best individuals provided by other techniques $s_3(g)$. When *TS* is enable, we have that more individuals are created using methods from *TS* for continuous optimization $s_1(g)$, then we have contribution of the best individuals in the actual generation $s_2(g)$, and individuals from other optimization techniques $s_3(g)$. This is a deterministic procedure, and the quantities $s_1(g)$, $s_2(g)$, and $s_3(g)$ can be modified by *AIIS*. If *AIIS* decides, we can include some of the individuals store in *TL*.
10. Increase pheromone level of selected parents.
11. Repeat steps 11 to 17 until we generate NMAX successors. The size of each new population is variable, as well as the number of parents selected for mating. In *HEM* we are mimicking human evolution, where successors do not kill their parents, and this must be a default condition, but this condition can change eventually if evolution decides via their adaptive intelligent/intuitive inference system a different situation. This can be controlled using a special genetic effect for this situation.
12. Apply the recombination operator for obtaining an offspring (a new gr_i). This step is achieved in concordance of what we programmed in the genetic operator ge_{igen}; i.e. valid combinations are *M-F*, *F-M*, and *0* for bisexual recombination.

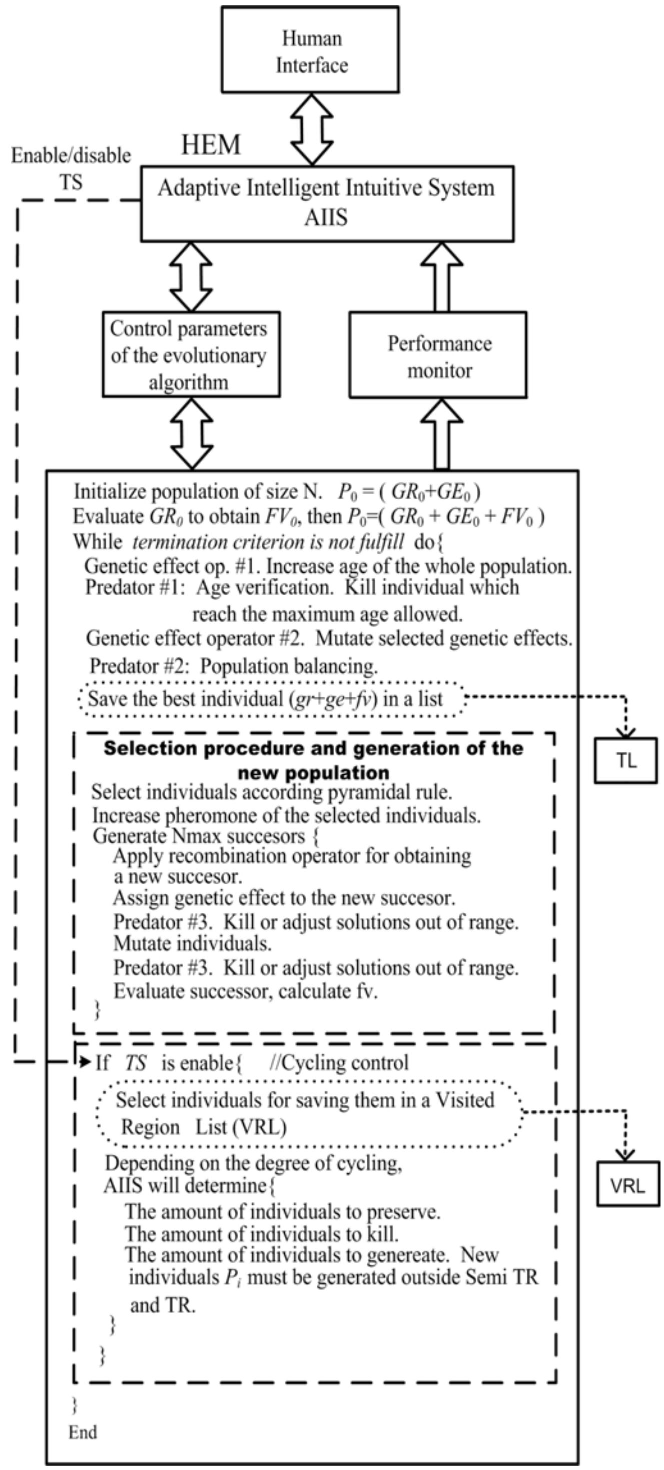

Fig. 2: General structure of *HEM*.

13. Assign to this new offspring gr_i their corresponding genetic effects, some attributes can be set to a default value, but other attributes like ge_{igen} must be set randomly.
14. "Predator #3" is an operator dedicated to eliminate all individuals that lays out a valid landscape. This operator can be programmed in two ways: one is to kill all individuals with parameter's values gr_i that are out of a valid range, this is a very drastic solution. A milder solution is to adjust the parameter values gr_i, modifying only the parameters that are

out of range, and this is achieved assigning the maximum value allowed in the corresponding frontier.

15. In this step we apply to the individual (only to gr_i) the mutation operator.
16. Apply the operator "Predator # 3".
17. Calculate the fitness value (fv_i) for the actual individual.
18. If *AIIS* disables the flag *TS*; i.e. *TS* = 0, then continue with step 20. Otherwise, *AIIS* will begin its respective task consisting in determining the degree of cycling; using this mechanism we can infer amounts of individuals to save in *VRL* [8], to preserve, to create and to kill. A *VRL* is defined as $VRL = \{(\zeta_i, \rho_i, \varphi_i)\}_{i=1}^{M}$, where M is the number of all listed visited regions; ζ_i is the center of a visited region, which is a sphere with radius ρ_i; and the frequency of visiting this region is represented by φ_i. We devoted section III of this paper to explain this part of *HEM*.
19. *AIIS* will determine the amount of new individuals to generate. Individuals should be created considering *TL* and *VRL*. With *TL* we can calculate Tabu Regions (*TRs*) where the points in *TL* are centers of spheres with radius r_{tr} >0. A semi-*TR* has a radio r_{STR} and it surrounds *TRs*. If a trial point lies in Semi-*TRs*, we will need to apply a special procedure which creates special neighborhood trial points for avoiding returning back to a vicinity of a previous visited solution.
20. Return to step 3.
21. This is the end of the program

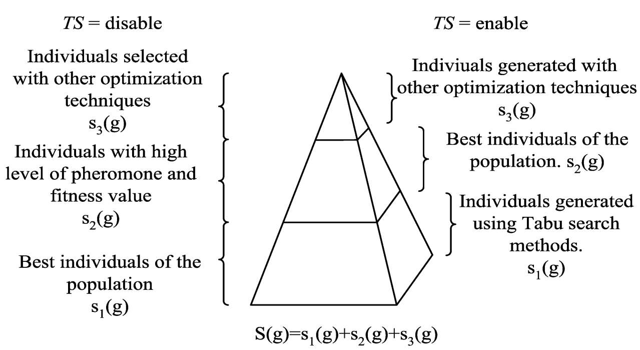

Fig. 3: Selection pyramidal rule. This pyramid is divided in three parts; each part represents the amount of individuals of each class that HEM is going to use for creating the next population. Here we can see that the best individual of the actual population are in the base of the pyramid, so the new population will have more individuals of this kind. AIIS can change the percentage value of each part of the pyramid.

III. Mediative fuzzy modeling

One task of *AIIS* in *HEM* is helping to reduce the number of iterations where the algorithm is being trapped in local minima. *AIIS* will detect the abovementioned situation rating in percentage terms the number of iterations that the algorithm is cycled around a fixed point. Cycling control is divided in two main parts: one is to put the already visited regions in a list (*VRL*); the second part is to use an intelligent mechanism to decide in the actual population, how many individuals we have to remove, how many individuals that are saved or included in *VRL* we have to preserve, the amount of individuals that we must eliminate, and the amount of individuals that we have to create outside *VRL*.

Since knowledge provided by experts can have big variations and sometimes can be contradictory, we are proposing to use a Contradiction fuzzy set to calculate a mediation value for solving the conflict. Mediative Fuzzy Logic is proposed as an extension of Intuitionistic fuzzy Logic [9,10].

An Intuitionistic fuzzy set A is given by

$$A = \{(x, \mu_A(x), \nu_A(x)) \mid x \in X\} \tag{2}$$

where $\mu_A : X \to [0,1]$ and $\nu_A : X \to [0,1]$ are such that

$$0 \le \mu_A + \nu_A \le 1 \tag{3}$$

and $\mu_A(x)$; $\nu_A(x) \in X$ denote a degree of membership and a degree of non-membership of $x \in A$ respectively

For each intuitionistic fuzzy set in X we have a "hesitation margin" $\pi_A(x)$, this is an intuitionistic fuzzy index of $x \in A$, it expresses a hesitation degree of whether x belongs to A or not. It is obvious that $0 \le \pi_A(x) \le 1$, for each $x \in X$.

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x) \tag{4}$$

Therefore if we want to fully describe an intuitionistic fuzzy set, we must use any two functions from the triplet [10].

1. Membership function
2. Non-membership function
3. Hesitation margin

The application of intuitionistic fuzzy sets instead of fuzzy sets [11,12,13], means the introduction of another degree of freedom into a set description, in other words, in addition to μ_A we also have ν_A or π_A. Fuzzy inference in intuitionistic has to consider the fact that we have the membership functions μ as well as the non-membership functions ν. Hence, the output of an intuitionistic fuzzy system can be calculated as follows:

$$IFS = (1-\pi)FS_{\mu} + \pi FS_{\nu} \tag{5}$$

where FS_{μ} is the traditional output of a fuzzy system using the membership function μ, and FS_{ν} is the output of a fuzzy system using the non-membership function ν. Note in equation (6) , when π =0 the *IFS* is reduced to the output of a traditional fuzzy system, but if we take into account the hesitation margin of π the resulting IFS will be different.

In similar way, a contradiction fuzzy set in X is given by:

$$\zeta(x) = \min(\mu_A(x) + \nu_A(x)) \tag{6}$$

where $\mu_A(x)$ represents the membership functions, and for the variable $\nu(x)$ we are using the term non-agreement instead no-membership, because we think this name is more adequate when we have contradictory fuzzy sets

The output is calculated using

$$MFS = \left(1 - \pi - \frac{\zeta}{2}\right) FS_{\mu} + \left(\pi + \frac{\zeta}{2}\right) FS_{\nu} \quad (7)$$

In this case, when the contradictory index ζ is equal to cero, the system's output can be reduced to an intituionistic fuzzy output or, in case that $\pi = 0$, it can be reduced to a traditional fuzzy output.

IV. Experimental Results

The experiments we are showing in this paper were developed for *HEM*. These experiments are focusing in reducing the problem of being trapped by local optimum, more specifically in the part of inferring the amount of individuals to create, to eliminate (kill), and to preserve. We used Sugeno Inference system to calculate FS_{μ} and FS_{ν}, so the system is divided in two main parts: the inference system of the membership function side, and the inference system of the non-agreement function side. For the first one, in Fig. 4 we are showing the membership functions Small, Medium and Large. They are used in a Sugeno Inference System, which in turn have three constant type outputs: FSCreate, FSKill and FSPreserve. For the output FSCreate we have the values: Nothing=0, Little=0.5, and Many=1; For the output FSKill we have: Nothing=0, Little=0.5, All=1; and the output Preserve has the values of: Nothing=0, More or Less=0.5, All=1. At the side of non-agreement functions, we have three fuzzy sets: NoSmall, NoMedium, and NoLarge, they are shown in Fig. 5. They are applied to a Sugeno Inference System with three outputs: nCreate, nKill, and nPreserve. For nCreate we have the constant values: Nothing=0, Little=0.5, Many=1. For nKill we have: Nothing=0, Little=0.5, and All=1. For nPreserve we have: Nothing=0, More or Less=0.5, and All =1. Using the membership function and the non-agreement functions we obtained the hesitation fuzzy set (Fig. 6) and the contradictory fuzzy set (Fig. 7). Finally, the Mediated fuzzy outputs Create given by equation (8) will infer the amount of individual to create outside *VRL*, Kill given by equation (9) will infer the amount of individuals in actual population that will be eliminated, using equation (10) we can calculate the amount of individuals to preserve in the actual population (see Figs. 8, 9 and 10).

$$FSCreate = \left(1 - \pi - \frac{\zeta}{2}\right) FS_{Create} + \left(\pi + \frac{\zeta}{2}\right) FS_{nCreate} \quad (8)$$

$$FSKill = \left(1 - \pi - \frac{\zeta}{2}\right) FS_{Kill} + \left(\pi + \frac{\zeta}{2}\right) FS_{nKill} \quad (9)$$

$$FSPreserve = \left(1 - \pi - \frac{\zeta}{2}\right) FS_{\Pr eserve} + \left(\pi + \frac{\zeta}{2}\right) FS_{n \Pr eserve} \quad (10)$$

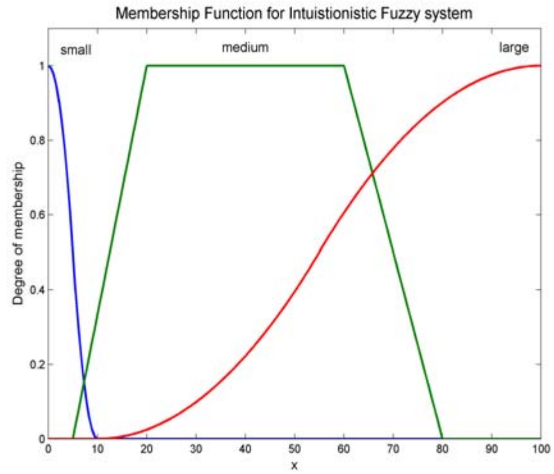

Fig. 4: Membership functions.

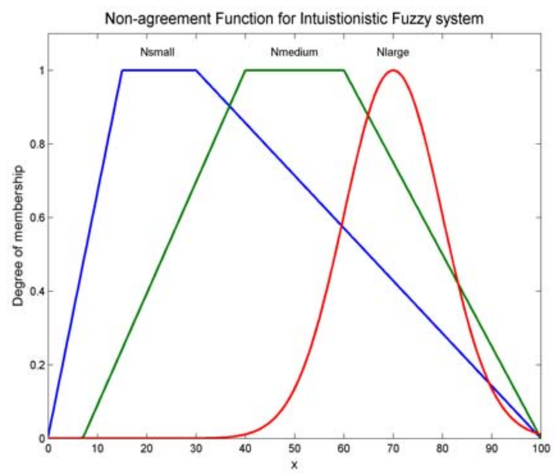

Fig. 5: Non-agreement membership functions.

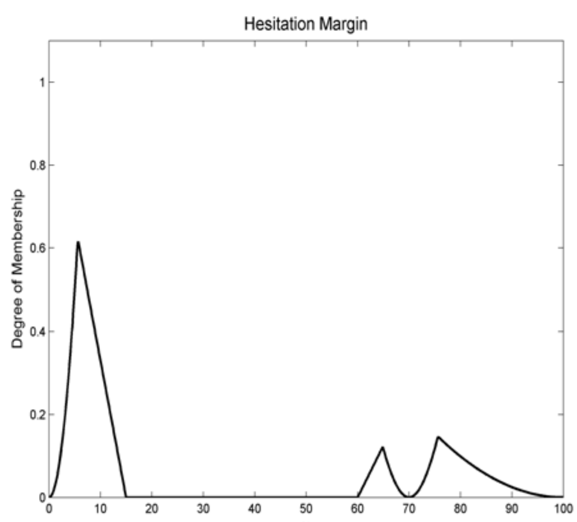

Fig. 6: Hesitation fuzzy set.

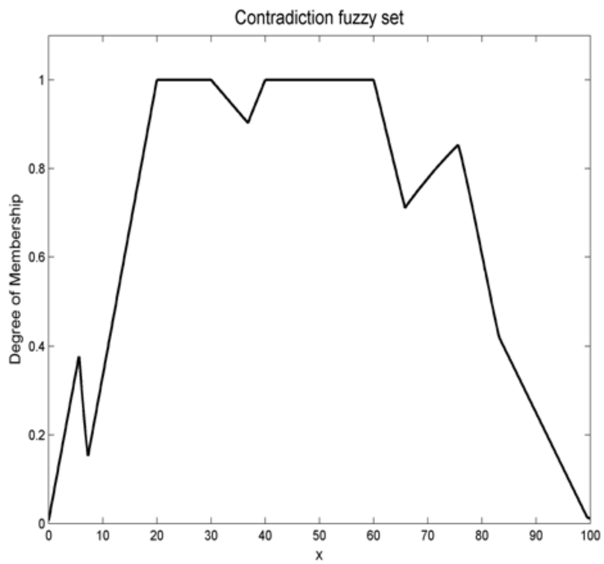

Fig. 7: Contradiction fuzzy set.

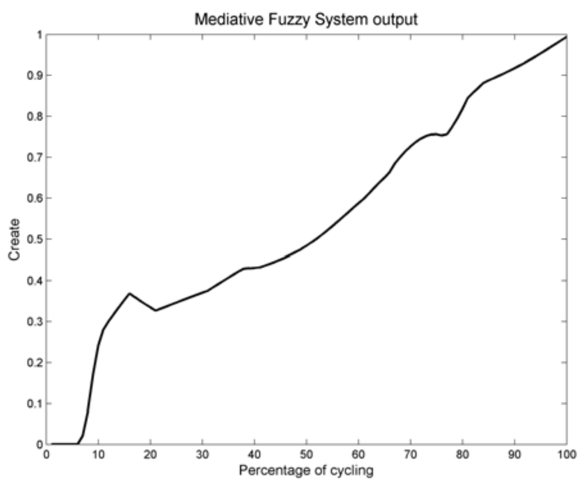

Fig. 8: Mediative Fuzzy System of the Create output.

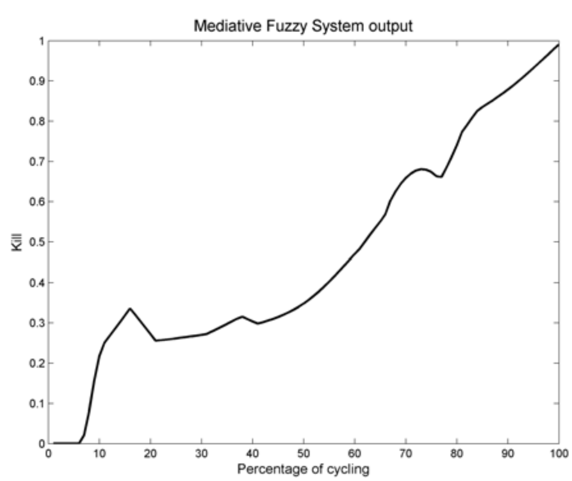

Fig. 9: Mediative Fuzzy System of the Kill output.

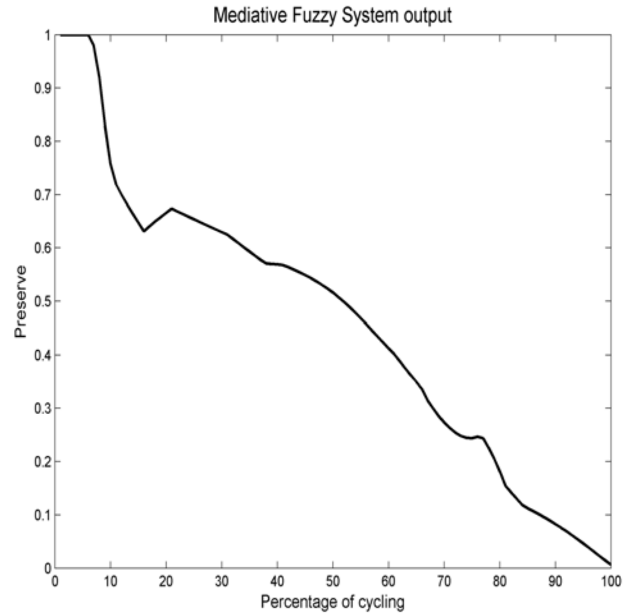

Fig. 10: Mediative Fuzzy System of the Preserve output.

V. Conclusions

It is a fact that the rigid form of static parameters commonly used in an evolutionary algorithm to perform its search and optimization task contradicts the dynamic nature of evolutionary algorithms, so it is beneficial that the algorithm evolves with the problem to provide a better solution. Self-adaptation is a concept which makes evolutionary algorithms flexible and closer to natural evolution. HEM is a versatile evolutionary algorithm because it incorporates the human expertise in a special system named AIIS, it is a self-adaptive algorithm, and allows the user to interact on-line with it. The method for reducing the cycling problem has been developed for HEM, but it can be applied to different evolutionary algorithms since it has several desirable advantages; evidently, the most important is to avoid getting trapped in local optima, so we used a Tabu list of the already visited regions in combination with the creation, elimination and preservation of individuals of the actual population. The last part, concerning population control was implemented using an innovative concept that we called Mediative Fuzzy Logic (MFL) because it is able to deal with traditional fuzzy sets, as well as with imperfect knowledge, this is knowledge with hesitation, and contradiction. We consider that this last part is very important because Mediative Fuzzy Logic is able to handle contradictory knowledge about the same issue provided by experts, and if there is no contradiction, or any doubt in the knowledge, MFL automatically is reduced to intuitionitistic or to traditional Fuzzy Logic. At present time we are still testing HEM, applying the above concepts we have promissory results.

REFERENCES

[1] S. G. Ficici, *Soluion Concepts in Coevolutionary Algorithms*. Ph.D Dissertation, Brandeis University, May 2004. (Computer Science Department Technical Report CS-03-243).

[2] E. K. P. Chong and S. H. Żak An Introduction to Optimization, Second Edition. Ed. Wiley, 2001.

[3] J.-S.R. Jang, C.-T. Sung, E. Mizutani, Neuro-Fuzzy and Soft Computing. A Computational Approach to Learning and Machine Intelligence. Prentice Hall, NJ, USA, 1997.

[4] O. Nelles, Nonlinear System Identification: From Classical Approaches to Neural networks and Fuzzy Models. Springer-Verlag Berlin Heidelberg. Germany. 2001. pp. 15, 457-511.

[5] D. Kalyanmoy, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, LTD. Great Bretain, 2002.

[6] O. Montiel, O. Castillo, A, Rodríguez, P. Melin, R. Sepúlveda, *The Human Evolutionary Model: A new approach for solving nonlinear optimization problem.* Paper accepted in JCIS 2005, 8TH Joint Conference on Information Sciences, Salt Lake City, Utah, USA. July 21-26, 2005.

[7] "Self adaptive software", December, (1997). DARPA, BAA 98-12, Proposer Information`Pamphlet, Available: www.darpa.mil/ito/Solicitations/PIP_9812.html.

[8] A.-R. Hedar and M. Fukushima, Tabu Search directed by direct search methods for nonlinear global optimization. Preprint submitted to Elsevier Science, 2004.

[9] K Atanassov, *Intuitionistic Fuzzy Sets: Theory and Applications*, Springer-Verlag, Heilderberg, Germany, 1999.

[10] O. Castillo and P. Melin, *A New Method for Fuzzy Inference in Intuitionistic Fuzzy Systems*, Proceedings of the International Conference NAFIPS 2003, IEEE Press, Chicago, Illinois, USA, Julio 2003, pp. 20-25.

[11] L. A. Zadeh, *Fuzzy Sets*, Information and Control, 8, 1965, pp. 338-353.

[12] L. A. Zadeh, *Quantitative Fuzzy Semantics*, Information Sciences, 3, 1971, pp. 159-176.

[13] G. Chen and T. T. Pham, *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*, CRC Press, Boca Raton, Florida, USA, 2001, pp. 119-144.

[14] L.-X. Wang, A Course in fuzzy systems, and control, Prentice Hall, USA, 1997.

Oscar Montiel is a Researcher with the Center for Research in Digital Systems in Tijuana, Mexico. He has published 5 Journal papers and 15 Conference Proceedings papers. He is member of IEEE. He is founding member of the IEEE Computational Intelligence-Chapter Mexico, and member of the Hispanic American Fuzzy Systems Association. He is also member of the International Program Committees of several Conferences, and reviewers of several International Journals. At the moment, he is currently finishing his Ph.D. Dissertation on evolutionary algorithms and fuzzy logic at UABC University.

Oscar Castillo is a Professor of Computer Science in the Graduate Division, Tijuana Institute of Technology, Tijuana, Mexico. In addition, he is serving as Research Director of Computer Science and head of the research group on fuzzy logic and genetic algorithms. Currently, he is President of HAFSA (Hispanic American Fuzzy Systems Association) and Vice-President of IFSA (International Fuzzy Systems Association) in charge of publicity. Prof. Castillo is also Vice-Chair of the Mexican Chapter of the Computational Intelligence Society (IEEE). Prof. Castillo is also General Chair of the IFSA 2007 World Congress to be held in Cancun, Mexico. He also belongs to the Technical Committee on Fuzzy Systems of IEEE and to the Task Force on "Extensions to Type-1 Fuzzy Systems". His research interests are in Type-2 Fuzzy Logic, Intuitionistic Fuzzy Logic, Fuzzy Control, Neuro-Fuzzy and Genetic-Fuzzy hybrid approaches. He has published over 50 journal papers, 5 authored books, 10 edited books, and 160 papers in conference proceedings.

Jose Soria is a Professor with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico. He has published 4 Journal papers and 5 Conference Proceedings papers. He is member of the Hispanic American Fuzzy Systems Association.

Antonio Rodriguez is a Professor with the Division of Graduate Studies and Research of UABC University, Tijuana, Mexico. He has published 4 Journal papers and 15 Conference Proceedings papers. He is member of IEEE. He is member of the IEEE Computational Intelligence-Chapter Mexico, and member of the Hispanic American Fuzzy Systems Association.

Hector Arias is a Professor with the Division of Graduate Studies and Research in Tijuana Institute of Technology, Mexico. He has published 4 Journal papers and 5 Conference Proceedings papers. He is member of the Hispanic American Fuzzy Systems Association.

Roberto Sepulveda is a Researcher with the Center for Research in Digital Systems in Tijuana, Mexico. He has published 5 Journal papers and 15 Conference Proceedings papers. He is member of IEEE. He is founding member of the IEEE Computational Intelligence-Chapter Mexico, and member of the Hispanic American Fuzzy Systems Association. He is also member of the International Program Committees of several Conferences, and reviewers of several International Journals. At the moment, he is currently finishing his Ph.D. Dissertation on evolutionary algorithms and fuzzy logic.