The Human Evolutionary Model: A new approach for solving nonlinear optimization problems

Oscar Montiel[1] Oscar Castillo[2] Patricia Melin[2] Antonio Rodríguez[3] Roberto Sepúlveda[1]

[1]CITEDI-IPN. Av. del Parque #1310, Mesa de Otay, Tijuana, B.C., México. oross@citedi.mx, rsepulve@citedi.mx
[2]Department of Computer Science, Tijuana Institute of Technology. P.O. Box 4207, Chula Vista, CA 91909 USA. ocastillo@tectijuana.mx, pmelin@tectijuana.mx [3]FCQI-UABC, Tijuana, B.C. México. ardiaz@uabc.mx

Abstract

In this paper we are proposing the Human Evolutionary Model (*HEM*) for solving nonlinear optimization problems. This is a synergetic model that uses intelligence and intuition as main driving forces. It is combining some fundamental aspects of different methods. Nowadays, this paper is one of the first documents where *HEM* is proposed, we obtained promising results.

Keywords: *HEM*, optimization, nonlinear.

1. Introduction

Since optimization is a central point in any decision making problem, recently, it has received enormous attention as a direct consequence of the rapid progress in computer technology including the availability of powerful software [1]. Obtaining valid and accurate models is an essential routine task in many practical problems of control engineering and sciences [2]. *HEM* is an evolutionary computational method inspired in human evolution; it can be classified as a global nonlinear optimization method.

2. The Human Evolutionary Model

The main idea of this computational model is to combine synergetically diferent techniques for performing search and optimization tasks [3]. We defined *HEM* as an eight tuple

$$HEM = (AIIS, P, O, S, E, L, TL, VRL)$$

where *AIIS* is the Adaptive Intelligent Intuitive System, *P* is the population of human like individuals, *O* is expressing a single or a multiple objective goals, *S* is the evolutionary strategy used for reaching the objectives expressed in *O*. *E* is the environmental, here we can have predators, etc., *L* is the landscape, i.e., the scenario where the evolution must be performed. *TL* is a Tabu List formed by the best solution found, *VRL* is a Visited Region List [4]. The population *P* has *N* individuals. Fig. 1 is a schematic representation of one individual which is comprised of 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.

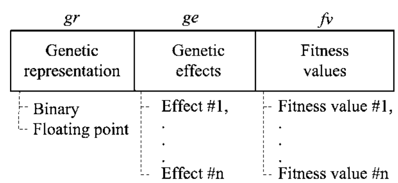

Fig. 1: Representing one individual in *HEM.*

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},...,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,…,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. 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.

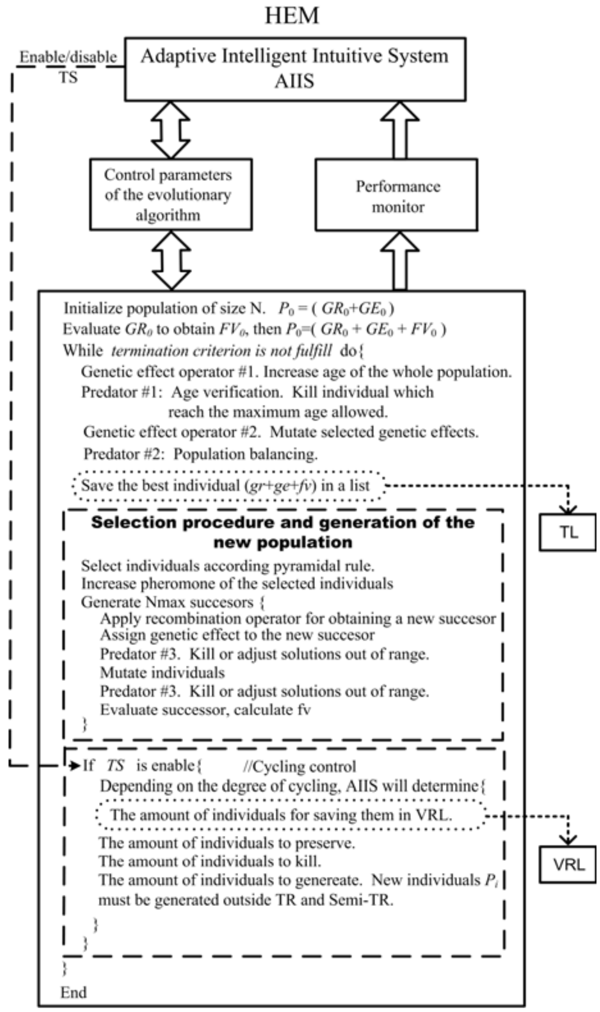

Fig. 2: General structure of *HEM*.

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.

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.

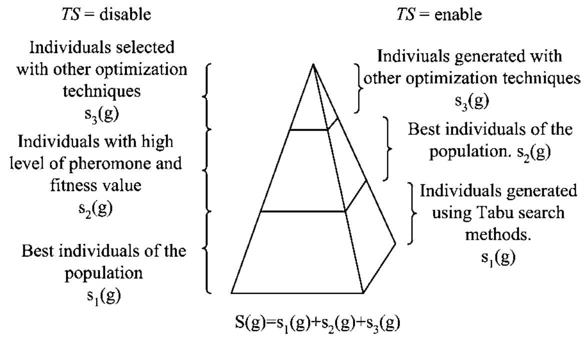

Fig. 3: Selection pyramidal rule.

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* [4], to preserve, 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.

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.

3. Experimental results

We performed several experiments using *HEM* for optimizing multimodal functions, and parameter estimation of static and dynamic systems. In this paper, we are showing results obtained in one of this test. In general, we had the next starting test conditions: the parameters to be optimized were coded using real numbers, we used the fuzzy recombination operator [5], and the discrete mutation (*DM*) operator normally use in the Breeder Genetic Algorithm (*BGA*); we used the genetic effect "gender", "actual age", "maximal age allowed", and "parent survival"; the maximum age allowed for each parent was "4"; we used the classical test function named Rastrigin, since this is a fairly difficult problem due to its large search space and its large number of local minima. In general, the Rastrigin function is defined by equation (1), where n is the dimension of the problem.

$$F(\vec{x}) = A \cdot n + \sum_{i=1}^{n} x_i^2 - A \cdot \cos(w \cdot x_i) \qquad (1)$$

The variables A and w control the amplitude and frequency modulation respectively. For this experiment we use A=10, $w=2\pi$, $n=2$, and $x_i \in [-50,50]$. Fig. 4 shows a plot of equation (1) for two dimensions, we used the best individual at each generation for drawing the searching path in order to locate the global minimum of this function. Fig. 5 is an enlarged view of Fig. 4, here we are emphasizing part of the behavior of the evolution of the worst individual throw generations. Note that although the whole population is converging to the global minimum, we do not have to worry about getting trapped in local minima since *HEM* has a special mechanism for escaping from these traps. In Fig. 6 we can appreciate better the best and worst searching paths obtained via evolution. In Fig. 7, we can see that although the experiment began with individuals with very low fitness, rapidly we had fitter individuals, so we had to use logarithmic scale. The task of Predators in *HEM* is helping to maintain a balance in population size, this is the case of Predator #1 which verifies the genetic effect $ge_{imaxAge}$ killing those individuals that exceed the value "maximum age allowed", in this case "4". This effect is shown in Fig. 8. Note, that the selection mechanism is the main method for removing individuals, even they are younger than $ge_{imaxAge}$.

4. Conclusions

HEM is an intelligent and intuitive evolutionary method that works synergistically taking advantage of other techniques; one can take an appropriated evolutionary methodology (GA, BGA, etc.) and adapt it to *HEM*. *HEM* through *AIIS* can change functional parameters, enable/disable features, in its more primitive form it will work as if the evolutionary algorithm selected were alone. An interesting result shown in this work is the reduction of population size focusing in individuals that are producing succesors with better fitness values, without the fear of being trapped by local minima since always there is available a special mechanism controlled by an intelligent/intuitive system avoiding local minimum using Tabu regions.

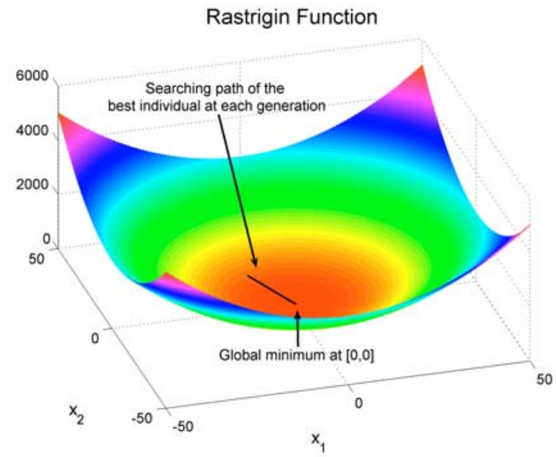

Fig. 4: Plot of the function in the range of [-50,50].

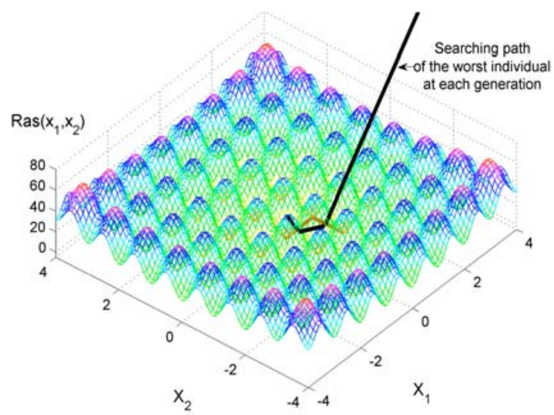

Fig. 5: Enlarged view of Fig. 4.

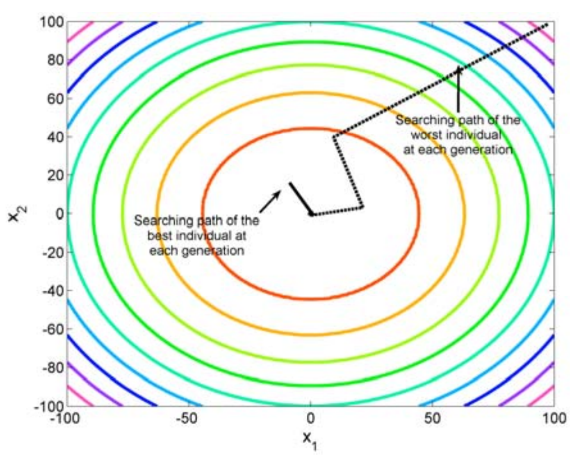

Fig 6: Searching paths of the best and the worst individuals.

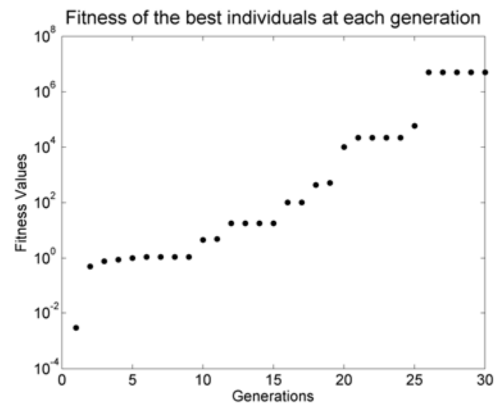

Fig. 7: We use logarithmic scale for plotting the fitness value of the best individual at each generation.

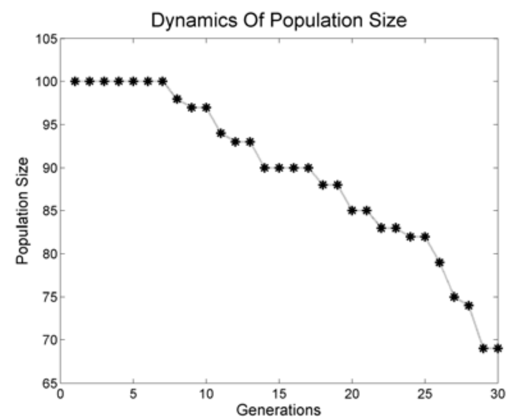

Fig. 8: HEM uses a bounded dynamic population size. In this experiment we used 100 individuals for the initial population, with 330 and 50 individuals as the upper and lower bound respectively.

5. References

[1] Edwin K. P. Chong and Stanislaw H. Żak, *An Introduction to Optimization, Second Edition.* Ed. Wiley, 2001.

[2] Lennart Ljung, *System Identification. Theory for the User. Second Edition.* Prentice Hall PTR, USA, 1999.

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

[4] Abdel-Rahman Hedar, Maso Fukushima, Tabu Search directed by direct search methods for nonlinear global optimization. Preprint submitted to Elsevier Science, 10 June, 2004.

[5] Voigt H.M., Mühlenbein, D. Cvetkovic (1995), “Fuzzy Recombination for the Breeder Genetic Algorithm”, Proceedings of the Sixth International Conference on Genetic Algorithms, published by Morgan Kaufmann, pp. 1104-111.