**Proceedings of the ASME 2011 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2011
August 28-31, 2011, Washington, DC, USA**

DETC2011-47584

GEO + ES HYBRID OPTIMIZATION ALGORITHM APPLIED TO THE PARAMETRIC THERMAL MODEL ESTIMATION OF A 200N HYDRAZINE THRUSTER

Roberto Luiz Galski
National Institute for Space Research
1758, Av. Dos Astronautas
São José dos Campos-SP-Brazil
551239456386
galski@ccs.inpe.br

Heitor Patire Júnior
National Institute for Space Research
1758, Av. Dos Astronautas
São José dos Campos-SP-Brazil
551239456245
heitor@dem.inpe.br

Fabiano Luis de Sousa
National Institute for Space Research
1758, Av. Dos Astronautas
São José dos Campos-SP-Brazil
551239456196
fabiano@dem.inpe.br

José Nivaldo Hinckel
National Institute for Space Research
1758, Av. Dos Astronautas
São José dos Campos-SP-Brazil
551239456200
hinckel@dem.inpe.br

Pedro Lacava
Aeronautical Technological Institute
50, Praça Marechal E. Gomes
São José dos Campos-SP-Brazil
551239475974
placava@ita.br

Fernando Manuel Ramos
National Institute for Space Research
1758, Av. Dos Astronautas
São José dos Campos-SP-Brazil
551239456024
fernando@lac.inpe.br

ABSTRACT

In the present paper, a hybrid version of the Generalized Extremal Optimization (GEO) and Evolution Strategies (ES) algorithms [1], developed in order to conjugate the convergence properties of GEO with the self-tuning characteristics present in the ES, is applied to the estimation of the temperature distribution of the film cooling near the internal wall of a thruster. The temperature profile is determined through an inverse problem approach using the hybrid. The profile was obtained for steady-state conditions, were the external wall temperature along the thruster is considered as a known input. The Boltzmann's equation parameters [2], which define the cooling film temperature profile, are the design variables. Results using simulated data showed that this approach was efficient in recuperating those parameters. The approach showed here can be used on the design of thrusters with lower wall temperatures, which is a desirable feature of such devices.

1. INTRODUCTION

In the last 20 years, a considerable number of global optimization methods have been developed. Most of them are based on natural phenomena analogies, trying to copy the efficiency and simplicity of observed self-optimized processes in nature. Algorithms based on the evolution of species [3,4], on the annealing of metals [5], on the functioning of the brain [6], on the immune system [7] and even on the social behavior of ants [8] have been developed and used to get optimized solutions for many science and engineering problems. Among them, perhaps the most commonly used are simulated annealing (SA) [5], genetic algorithms (GA) [3] and their derivatives.

The Generalized Extremal Optimization algorithm (GEO) [9–11], like SA and GA, is a stochastic algorithm, but unlike these ones, it has only one free parameter (τ) to be set, instead of three or more.

GEO was developed as a generalization to the EO method [12] and, since then, has been successfully applied to a broad variety of science and engineering real-world problems [9-11,13-19]. It is a global search meta-heuristic, based on a model of natural evolution [20], and specially devised to be used in complex optimization problems. It has its fundaments on the Self-Organized Criticality (SOC) theory, which has been used to explain the power law signatures that emerge from many complex systems [21].

ES [22] is a well known technique, whose first developments remounts back to the early 1960's and whose further versions were among the first algorithms to include self-tuning (of their internal parameters) as a feature.

The GEO + ES hybrid algorithm [1] was developed as way of combining the good convergence properties of a real-valued version of the GEO algorithm with the self-tuning characteristics present in the ES methods. It also introduces a new mutation operator that allows the algorithm to change the locality of the search at each iteration. Three performance tests with test functions used by other authors [23-25] for testing their own algorithms were

also performed for GEO + ES in [1] and compared, all showing excellent performance for the GEO + ES hybrid.

Low thrust bipropellant rocket engines development present a difficult task regarding the heat load on the thruster wall. The adiabatic combustion temperature for stoichiometric mixture of most propellant pairs is much higher than the allowable temperature limit of the combustion chamber material. A compromise solution between the energetic efficiency (near stoichiometric mixture ratio) and wall heat load (which requires a fuel or oxidizer rich film near the wall) must be found.

The heat load on the thruster wall depends on the temperature of the combustion products in the gas film near the wall, and the heat transfer coefficient. The determination of these parameters by analytical methods is subject to very large errors due to difficulties in modeling the atomization, mixing and combustion processes inside the chamber. The experimental measurement of the heat transfer coefficient and near wall gas film temperature is also difficult to perform.

In this paper we describe a method to estimate the near wall gas film temperature from the measured outside wall temperature during the wall heating transient. The success of the method depends on the hypothesis that the aero-thermodynamic process inside the chamber is much shorter than the wall heating transient.

In [26], preliminary results were showed. In this article, the GEO + ES hybrid algorithm is thoroughly presented and used to solve the inverse problem of thermal model parameter estimation for the 200N hydrazine thruster being developed by the National Institute for Space Research of Brazil, INPE (in portuguese, Instituto Nacional de Pesquisas Espaciais). From the experimental data on the temperatures along the thruster external wall, and considering steady-state conditions, one wants to find the optimal estimates for the Boltzmann's equation parameters [2], e.g., the coefficients of the Boltzmann equation that best match the outside wall temperature profile during the initial thermal transient The temperatures of the burning mixture film on the vicinities of the thruster internal wall are defined with help of the Boltzmann equation. The importance of finding the optimal estimates is that, once accurately found, they allow precise computerized thermal simulations for the thruster.

2. GEO + ES HYBRID ALGORITHM

In the following, ES and GEO stand alone algorithms are described, being followed by the description of the GEO + ES hybrid.

2.1 ES algorithm

ES is an optimization technique based on the ideas of adaptation and evolution, being another member of the family of the Evolutionary Algorithms (EA) [22]. ES methods use real coded vectors (non binary) and mutation mainly, among others, as operators. As it is common in EA, the operators are applied in order: recombination, mutation, evaluation of the adaptation function and natural selection. Applying this loop one time is called a generation, and it is repeated until a stop criterion is reached.

The first ES methods were based on a population of only two individuals per generation: one search space point (the father) and one mutation of it (the son), being used the notation (1+1)-ES. More recent versions employ populations of fathers and sons, such that the notation ($\mu+\lambda$)-ES indicates that both populations take part of the natural selection and the notation (μ,λ)-ES indicates that only the sons population, λ, participates of the natural selection. On the ES, the selection pressure is rather high, since, typically, $\lambda>>\mu$. The mutation, in its simpler version, is obtained by adding to each design variable vector component values coming from the same Gaussian distribution. The size or intensity of this mutation, i.e., the standard deviation of the Gaussian distribution, usually vary during the search, evolving together with the design variables, in a process known as self-adaptation. The self-adaptation is perhaps the main contribution from ES methods to the optimization field.

A mutation operator version with ES, called one size uncorrelated mutation, uses only one Gaussian distribution, described by $\text{gauss}(0,\sigma) = \sigma\cdot\text{gauss}(0,1)$, as source of perturbation of size σ for all the N components of the vector $\mathbf{X}$ containing the N design variables. In this way, each vector $\mathbf{X}$ generated by the mutation operator has an associated σ. The σ parameter, by its turn, is considered as an additional variable by the ES. Then, each ES individual is composed by the set $\{\mathbf{X},\sigma\}$, and σ is mutated multiplying it by a variable with lognormal distribution with zero mean and standard deviation α. In this way:

$$\sigma^{n} = \sigma^{n-1} \cdot e^{\text{gauss}(0,\alpha)} \quad (1)$$

$$X_j^{n} = X_j^{n-1} + \text{gauss}_j(0, \sigma^{n}) \quad (2)$$

Where the subscripts n-1 and n are used just to signalize value before mutation and after mutation, respectively. In the Eq. 2 above, the subscript j of the function gauss(.) indicates that each design variable suffer a different mutation, despite coming from the same random variable (same probability distribution). It is important to notice that, despite each individual being a $\{\mathbf{X},\sigma\}$ pair, the adaptation function used in the selection process is the objective function, $F(\mathbf{X})$, of the optimization problem originally formulated and not a $F(\{\mathbf{X},\sigma\})$. The mutation mechanism described by the Eq.s 1 and 2 generates a self-adaptation for the mutation size σ, since the selection process acts, at each generation, on the $\mathbf{X}$ vectors resulting from the mutations done under several σ's, but only the best $\mathbf{X}$ vectors survive, carrying with them their respective σ's. The parameter α in the Eq. 1 becomes, then, the only free parameter of this ES version. This parameter is commonly called learning rate (of the ES about the $F(\mathbf{X})$). Basically, the parameter α defines how rapidly the value of σ can vary at each generation (iteration) of the ES. High values for α indicate that the ES can perform the search by rapidly passing from a global (high σ's) to a local search (low σ's) and vice-versa.

According to [22], the essential characteristics present in the ES are: (i) ES methods are, in general, used for optimizing problems with continuous design variables (real); (ii) Mutation is the main operator used to generate new individuals; (iii) Design variables mutation is implemented by adding noise coming from a Gaussian distribution and; (iv) The parameters that control the mutation are modified along the algorithm execution.

2.2 GEO algorithm versions

Since GEO's first appearance in the literature [9], it already had a variant called GEO_{var}. In this section, for conciseness, references to GEO must be understood as references to both GEO and GEO_{var}, unless explicitly stated otherwise.

In [1], four improvements for the canonical GEO algorithm [9] were suggested. One of them, called GEO_4, uses real coded variables instead of the binary ones used by the canonical GEO and has presented very good performance results with test functions, in terms of convergence properties. All GEO versions and ES share the characteristic of using mutation as the main operator to generate individuals. The GEO_4 version is the only one that shares with the ES also the characteristic of using real instead of binary codification of the design variables. Considering this and also considering its excellent performance, GEO_4 was chosen to form the GEO + ES hybrid.

In what follows, GEO canonical version and the real coded improved version GEO_4 are described.

2.2.1 The canonical GEO

The description of the canonical GEO presented in this subsection can also be found, for instance, in [10,11], including application examples and performance comparisons with other stochastic algorithms.

The GEO algorithm is a stochastic method, does not make use of derivatives and can be applied to non-convex or disjoint problems. It can also deal with any kind of variables, either continuous, discrete or integer. The τ free parameter allows one to set up the determinism degree of the search, from a random walk ($\tau = 0$) to a deterministic search ($\tau \to \infty$).

A flowchart for GEO and its variant GEO_{var} is presented in the Figure 1. In the flowchart, F(**X**) is the objective function, k is the ranking value of the bit, L_j is the number of bits of the design variable "j", and L with no subscript is the number of bits encoding all the design variables.

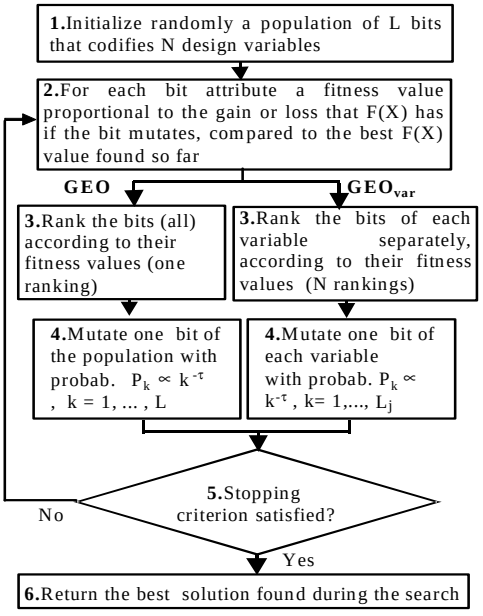

Figure 1. FLOWCHART FOR GEO AND GEO_{var}.

In GEO a string of L bits is considered a population of species. That is, each bit is a species. The string encodes the N design variables. For each of them is associated a fitness number that is proportional to the gain (or loss) the objective function value has in mutating (flipping) the bit. Regardless of the design variable it belongs to, all bits are ranked from 1, for the least adapted bit, to L for the best adapted. A bit is then mutated according to the probability distribution $P(k) \propto k^{-\tau}$, where k is the rank of a selected bit candidate to mutate, and τ is a free control parameter, set prior to execution. For $\tau \to 0$, any bit of the string has the same probability to be mutated, while for $\tau \to \infty$, only the least adapted bit can be mutated. The meaning of this is that for $\tau \to 0$, GEO performs a random walk in the binary discretized search space, whereas for $\tau \to \infty$, it performs only deterministic (best choice) movements. In practice, due to the exponential character of the exponential distribution of P(k), for values of $k > 10$ the probability that other bit than the least adapted be mutated is very low. After the bit is mutated, the procedure is repeated until a given stopping criterion is reached, i.e., a predetermined number of objective function evaluations, and the best configuration of bits (the one that gives the best value for the objective function) found is returned.

In a variation of the canonical GEO just described, called GEO_{var}, the bits are ranked separately for each sub-string that encodes each design variable, and N bits (one for each variable) are flipped at each iteration of the algorithm. In previous works, [9,10], it was observed that this implementation seems to be more efficient than the canonical one for cases in which the problem being tackled has only bound constraints (constraints that represent the limits for the design variables).

2.2.2 The GEO_4 version

This version introduces in GEO the internal representation of the design variables in the real domain, as well as, an alternative framework for the mutations. To the problems where the domain and the design variables original representation form are, respectively, the real domain and the decimal numeric system, using internally to the algorithm real instead of binary representation for the design variables is an advantage.

As seen in the preceding subsection, the canonical GEO uses internally binary codification to represent the design variables. However, there are studies pointing that the binary codification nor always is the most indicated form of representation [22]. Regarding this, Fig. 2 helps illustrating what happens. It shows a four bit tree discretization diagram for one single design variable, leading to a precision ε, as indicated in the figure. The vertical red lines crossing the x axis indicate movements between adjacent points on the discretized search space that are only achieved by simultaneous mutation (i.e., flipping) of two or more bits, as indicated by the numbers inside the dashed circles. Eventually, all the bits used in the discretization need to be changed, as happens on Fig. 2 for the points "0111" and "1000". This fact is a characteristic inherent to the binary codification system. However, as GEO performs only one bit mutation per variable on each algorithm generation (iteration), it means that the movements between adjacent points crossing the red lines are "forbidden" (impossible) for GEO within a single iteration.

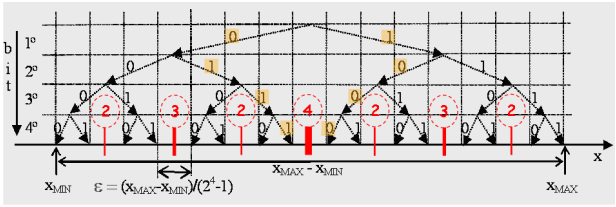

Figure 2. FOUR BIT DISCRETIZATION TREE AND THE "FORBIDDEN" MOVEMENTS FOR GEO WITHIN A SINGLE ITERATION.

Moreover, the canonical GEO changes the design variables by commuting the binary digits one by one during the fitness attribution phase. As the binary representation has base 2, it implicitly means that the mutation magnitudes are incremented by a multiplicative scale factor of 2, as illustrated on Fig. 3 for the point "0100". As can be seen in the Fig. 3, the fourth and less significant bit has a magnitude of ε, the third has 2ε, the second 4ε, and the first and most significant bit has a magnitude of 8ε, that is approximately equal to half the size of the search space, i.e., $(X_{MAX}-X_{MIN})/2$. Independently of the number of bits used for the discretization, this last fact remains true, i.e., the mutation of the most significant bit has a magnitude of $\sim 1/2(X_{MAX}-X_{MIN})$. It also means that the greatest mutation at each iteration as a fixed size equal to half of the search space of the respective design variable. It is reasonable to imagine that, for some objective functions, a different size for the greatest mutation could improve the efficiency of the algorithm.

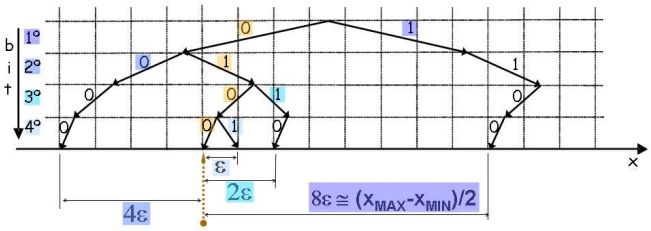

Figure 3. GEO MUTATIONS FOR THE POINT "0100".

All the aforementioned situations, described in the previous paragraphs, regarding the use of a binary representation with GEO were addressed in the reformulated version of the mutation framework that GEO_4 was born from. To the authors' knowledge, such framework was not proposed so far, and represents, in fact, a new mutation operator for AE, since it can be applied not only to GEO_4, but to AE in general (like GA, for instance). This new mutation operator is one of the main responsible for the convergence improvements obtained by GEO_4 over GEO with test functions [1].

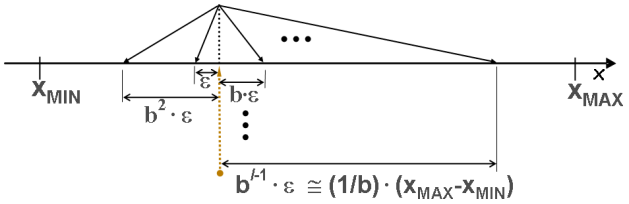

Figure 4. GEO_4 MUTATION FRAMEWORK.

Figure 4 presents the GEO_4 mutation framework, where l real-valued mutations are added to the real-valued design variable x. It is assumed that the present iteration value of x during a search is as indicated by the vertical dotted line. First of all, the sign of ε, the first (and smallest) mutation, is randomly chosen from an uniform distribution. Then, the sign of all the other mutations is chosen in a way to switch (be the opposite) regarding the preceding one. The scale factor among consecutive mutations is given by b, which is a real-valued parameter greater than 1. In the Fig. 4, the case where the sign of the first mutation happened to be negative and an even number of mutations l is shown. For GEO_4, given the values of b and l, the precision ε of the design variable x is calculated by:

$$\varepsilon = \frac{(X_{MAX} - X_{MIN})}{(b^l - 1)} \tag{3}$$

In the canonical GEO, the precision is defined by the number of bits used. In GEO_4, besides the number of mutations, l, the precision is defined also by b, which becomes an additional parameter of the algorithm that needs to be set. This can be considered a drawback. The counterpart is that, now, the locality of the search performed by GEO_4 at each iteration is adjusted prior to execution with the help of b. As indicated in Fig. 4, the greater mutation has a size (or magnitude) of $b^{l-1}\varepsilon$. When $b \rightarrow 1$, $\varepsilon \rightarrow \infty$ (please see Eq. 3), and the mutation size also tends to ∞, meaning very low locality for the search. When $b \rightarrow \infty$, ε and the mutation size tend to zero, meaning a search that is highly local. Mutations greater than 100% of $(X_{MAX}-X_{MIN})$ in size would mean side restriction violation of the design variables. To avoid this, GEO_4 replaces any mutation that would cause such situation by a totally random mutation for x within $[X_{MAX},X_{MIN}]$. A very big size mutation means very low locality during a search. A fully random one has also very low locality. In this way, the mentioned replacement helps avoiding a problem while produces practically the same effect.

The flowchart of Fig. 1 is still valid for GEO_4 and GEO_{var4}, but references to "bit" must be understood as "mutation". Also, the mutations cited in step 2 of the flowchart must be performed as explained in the two preceding paragraphs.

2.3 GEO + ES hybrid algorithm

In the ES, the parameter σ defines the size of the mutations that affect **X** and, as a consequence, defines also the locality of the search. In GEO_4, as just seen, the locality of the search is defined by the parameter b. This way, by analogy to the ES, the idea is to apply to the parameter b a variation mechanism similar to that used in the ES for the parameter σ. However, preliminary tests using function tests [1] have shown that the best way of mutating b is not by multiplication of a random variable with lognormal distribution, as indicated by Eq. 1, but by the addition of a random variable with a Gaussian distribution. Then:

$$b^n = b^{n-1} + \text{gauss}(\delta, \alpha) \tag{4}$$

Where the subscripts n-1 and n signalize value before mutation and after mutation, respectively.

Besides the learning rate, α, Eq. 4 presents yet the parameter δ, that is the mean of the Gaussian distribution used to mutate b. It indicates the bias imposed in the mutation of b. If δ=0, there is no bias. Imagining a search starting with $b=b_{MIN}$ (>1), then, using δ>0 generates, to the end of many generations, a schedule with stochastically increasing values for b, even that not monotonic ones (except if α=0). Remembering that low values for b impose a sparse

search and high values for b impose a local search, then a scheduling with increasing values for b means a search that starts sparse, when $b \sim b_{MIN}$, and that ends local, when $b \gg 1$. The idea behind using $\delta \neq 0$ is to allow a better tuning flexibility to the algorithm at the expenses of tuning one more algorithm parameter. In the cases where this flexibility is not wanted or needed, it is just to use the same idea adopted by the vast majority of the ES, i.e., to set $\delta = 0$.

Regarding the way of muting **X**, the same sistematic already described for GEO_4 is used.

An important change introduced in GEO + ES, regarding GEO_4, is the use of selection by elitism instead of stochastic selection with τ. Selection by elitism, in case of GEO_4, implies in only selecting the best among all the L individuals generated in each iteration, not requiring, this way, any additional parameter.

The steps of the GEO+ES hybrid used in this paper are given by:

1. Initialize randomly and with uniform distribution between $\mathbf{X_{min}}$ and $\mathbf{X_{max}}$ a vector **X** containing the N design variables. Calculate the value of the objective function $F(\mathbf{X})$, do $\mathbf{X_{best}} = \mathbf{X}$ and save $\mathbf{X_{best}}$ and $F(\mathbf{X_{best}})$.
2. Define the number of mutations l_j, $j \in \{1,2,...,N\}$ for each variable X_j, such that $\sum_j l_j = L$, being L the total number of mutations acting on the N design variables. Define values for the parameters δ e α. Define values for the limits inferior, b_{MIN}, and superior, b_{MAX}, of the base, with $b_{MIN} > 1$. Do $b = b_{MIN}$.
3. Calculate the vector $\boldsymbol{\varepsilon}$, where $\varepsilon_j \equiv (X_{MAX_j} - X_{MIN_j})/(b^{l_j} - 1)$ and j is the variable index, i.e., $j \in \{1,2,...,N\}$. The element ε_j defines the mutation resolution of the j-th design variable. By resolution understand the least value to be added or subtracted from X_j.
4. Do $F(\mathbf{X})_{ref} = F(\mathbf{X})$, $F(\mathbf{X_{best}})_{ref} = F(\mathbf{X_{best}})$.
5. For each design variable $j \in \{1,2,...,N\}$ of vector **X**, do:
   a. Draw from uniform distribution a value for $c \in \{0, 1\}$.
   b. For each mutation $i \in \{1,2,..., l_j\}$ of the variable X_j, do:
      1st. Calculate the mutation size $m = b^{(i-1)} \cdot \varepsilon_j$.
      2nd. Calculate the mutation sign $s = (-1)^{(i-c)}$, where c is the binary value obtained in the step 5.a.
      3rd. Mutate X_j. First, do $X_{aux} = X_j$. Then, do $X_j = X_j + s \cdot m$, generating a mutated vector $\mathbf{X_i}$. Verify the limits: If $X_j > X_{MAX_j}$ or $X_j < X_{MIN_j}$, then draw a new $X_j \in [X_{MIN_j}, X_{MAX_j}]$ with uniform distribution and do $m = abs(X_j - X_{aux})$ and $s = (X_j - X_{aux})/m$.
      4th. Calculate the objective function value $F(\mathbf{X_i})$. Atribute to the mutation i na adaptation value $\Delta F(\mathbf{X_i}) = F(\mathbf{X_i}) - F(\mathbf{X_{best}})_{ref}$, that indicates the gain or loss the objective function has if the mutation i occurs, when compared with the best objective function value found up to the previous iteration. Next, if $F(\mathbf{X_i}) < F(\mathbf{X_{best}})$ then do $F(\mathbf{X_{best}}) = F(\mathbf{X_i})$ and $\mathbf{X_{best}} = \mathbf{X_i}$.
      5th. Return **X** to its non mutated condition: do $X_j = X_j - s \cdot m$.
   c. Choose, within the l_j mutations generated in the step 5.b, the one with the minor $\Delta F(\mathbf{X_i})$. Save the respective X_j at the j-th element of the vector $\mathbf{X_c}$.
6. Perform the mutations of the present iteration: do $\mathbf{X} = \mathbf{X_c}$, where $\mathbf{X_c}$ is the resulting vector from the execution of the step 5.c for each variable $j \in \{1,2,...,N\}$.
7. Verify the base adaptation: Calculate the new $F(\mathbf{X})$ value. If $F(\mathbf{X}) \geq 0.99 \cdot F(\mathbf{X})_{ref}$ then do $b = b + y$, with $y = gauss(\delta,\sigma)$, where $gauss(\delta,\alpha)$ is the value drawn from an uniform distribution with mean δ and standard deviation α. Verifique os limites da base: If $b > b_{MAX}$ or if $b < b_{MIN}$, do $b = b_{MIN} + gauss(0,1) \cdot (b_{MAX} - b_{MIN})$.
8. Repeat the steps 3 to 7 until a given stop criterion be satisfied.
9. Return $\mathbf{X_{best}}$ and $F(\mathbf{X_{best}})$.

When GEO + ES is compared to the canonical GEO, the main differences can be summarized by:

- Real valued vector instead of binary chain is used to represent the design variables;
- Mutations on the design variables are done by magnitudes (any base) instead of bits (base 2);
- Number of mutations is dissociated from the numeric precision of the design variables;
- Absence of the τ parameter: Selection is performed by elitism, instead of stochastically;
- Adaptedness of the base b, which controls both the locality and the stochasticity of the search;
- Existence of three adjusting parameters: α, the learning rate, δ, the learning bias, and l_j, the number of mutations.

3. THE 200N HYDRAZINE THRUSTER

The 200N thruster in development at INPE is intended for use in the apogee acceleration block of geostationary satellites and launch vehicle roll control. Fig. 5 gives a view of the thruster.

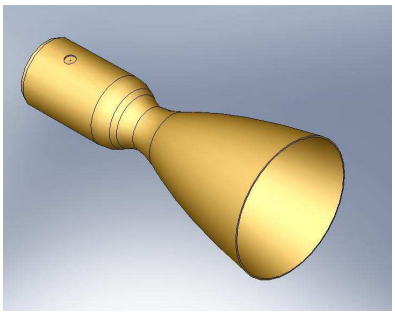

Figure 5. PERSPECTIVE VIEW OF THE 200N THRUSTER.

The thruster wall is manufactured in Inconel 600 and has a working temperature limit of 1300 K. The total length of the thrust chamber is 210mm. The combustion chamber is cylindrical with an internal diameter of 42mm.

The fuel is monomethyl hydrazine and the oxidizer is nitrogen tetroxide. The O/F mixture ratio is varied on the range of 0.6 to 1.3. The adiabatic combustion temperature for these mixture ratios goes from 2000 K to 2500 K. For an O/F mass ratio of 1 the adiabatic temperature of the combustion products for this propellant pair is 2360 K. The thruster is fire-tested in a test stand that includes

altitude simulation. The outer wall temperature is measured in a discreet number of points with type K thermocouples. An infrared camera has been used to monitor the entire external wall temperature.

Knowing the heat load on the thruster chamber wall is very important in order to establish a safe range of operating parameters of the thruster regarding mixture ratio, amount of fuel used in the film cooling and mechanical properties of the wall material at high temperature. The thermal model presented in the next section and the hybrid optimization algorithm described in the previous section have been used to simulate this temperature using thermal balance and comparing the external wall temperature simulated with the measured in the infrared camera.

4. THERMAL MATHEMATICAL MODEL

The thermal protection mechanism for the thruster wall is a film cooling on the inside and thermal radiation on the outside of the wall. To be able to determine with good precision the heat load on the inside of the wall one must know the temperature of the gas layer near the wall. The heat transfer convective coefficient is determined in the internal surface by Bartz`s equation [27]. The direct measurement of these quantities, or its determination from analytical models, is very difficult to be done.

Thermal models of thruster are usually constructed using a lumped parameter network formulation [28]. In this method, the thruster is divided into a number of lumped masses, called nodes, which are assumed isothermal. A thermal network is drawn connecting the nodes. A governing thermal energy expression is written for each node, resulting in a system of coupled equations whose solution yields the temperature of nodes. For the 200N hydrazine thruster, the thermal model was constructed using INPE´s PCTER thermal software package [29].

To apply the optimization technique, a lumped parameter model was constructed for the thruster using 160 nodes, as shown in Figure 6. As the thruster has radial symmetry along the longitudinal axis, only half of the longitudinal section needs to be modeled and the resulting model is bidimensional.

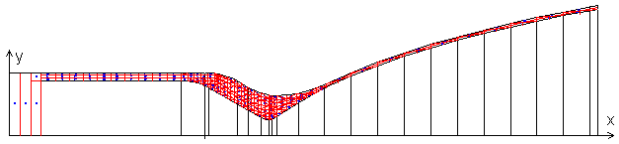

Figure 6. GEOMETRIC DISTRIBUTION OF THE NODES.

Heat from the hot gases inside the thruster is exchanged by radiation and convection with the thruster's internal wall, then is transferred by conduction to the external surface of the thruster and then to space by radiation. (external boundary conditions temperature).

Using the lumped parameter representation [28] and assuming steady state conditions, the heat balance at each one of the nodes takes the form of the following system of equations:

$$\overbrace{k\frac{dT}{dx}+k\frac{dT}{dy}}^{(1)}+\overbrace{\sum_{j=1}^{n+1}R_{ji}\sigma\varepsilon\left(T_j^{\,4}-T_i^{\,4}\right)}^{(2)}+\overbrace{h_i dT_i}^{(3)}=0 \qquad (5)$$

Where term (1) in Eq. 5 is the conductive term and the (2) is the radiative term and (3) is the convective term.

As is usual in this type of thruster, the propellant is injected from the periphery of the injector plate, as a way to create a thin cooling propellant film on the inner surfaces of the thruster, protecting them. The equation of Boltzmann [2] is the most used way of empirically determining what the resulting temperature of the cooling film is. It is given by:

$$y=\frac{A_1-A}{1+e^{(x-x_o)/dx}}+A_1 \qquad (6)$$

Where y is the film temperature, x is the longitudinal distance along the thruster, x_0 locates the smallest transversal section of the thruster (the bottleneck's position), dx is the longitudinal distance from the point where the film cooling temperature goes from level A to level A_1. A is the entrance fuel temperature, and A_1 is the adiabatic temperature of the combustion products.

When the set of parameters $\{A,A_1,x_0,dx\}$ is known, the film cooling temperature profile is obtained and the external wall temperature can be calculated using the direct model. When the set of parameters $\{A,A_1,x_0,dx\}$ is not known, but experimental data about the temperatures on the outer surface of thruster is available, then it is possible to formulate an optimization problem in which $\{A,A_1,x_0,dx\}$ are the unknowns and the objective is to find the values of them that lead to the best match between the observed data (measured temperatures) and the corresponding data obtained by solving Eq. 5. This is the inverse problem. In this paper, INPE PCTER thermal software package is used for solving Eq. 5 [29].

5. OPTIMIZATION PROBLEM FORMULATION

Mathematically, the optimization problem described in the last paragraph of the preceding section is stated as follows:

$$\textit{Minimize} \quad F(\mathbf{X})=\left\|\mathbf{T_S}(\mathbf{X})-\mathbf{T_D}\right\|_2 \qquad (7)$$

$$\textit{Subject to}: \quad \mathbf{X_{MIN}} \le \mathbf{X} \le \mathbf{X_{MAX}}$$

Where the objective function, F(**X**), is the Euclidian norm of the difference vector between the calculated and the given temperatures on the external surface of the thruster wall. $\mathbf{T_S}$ is the given temperature profile and $\mathbf{T_D}$ is the calculated one. The vector **X** is the design variables vector, i.e., $\mathbf{X} = [A,A_1,x_0,dx]$, that must remain between the side limits $\mathbf{X_{MIN}}$ = [1.0, 1500.0, 0.19, 0.005], and $\mathbf{X_{MAX}}$ = [40.0, 2400.0, 0.57, 0.3].

6. RESULTS

In order to evaluate the performance of the GEO + ES hybrid in the solution of the optimization problem described in section 5, the set $\{A,A_1,x_0,dx\}=\{23.0, 2087.0, 0.41, 0.238\}$ was considered and experimental data was synthetically generated using the PCTER software, as real experimental data on the outer wall temperature was not available. In this way, the optimal solution is known beforehand.

Three mutations per variable were used, so $l_j=l=3$ and L=12. The limits for varying b were set to b_{MIN}=1.05 and b_{MAX}=10. The values of δ=0.0 and of α=0.3 were used. After that, GEO + ES was run 10 times, each one with a different and random starting point. For each run, the limit of $5\cdot10^4$ evaluations of F(**X**) was used as the stopping

criterion. Each run took approximately 3.6 hours on an AMD Athlon (1.1GHz) PC computer with 896MB of RAM memory.

The ten solutions found with the help of GEO + ES plus the Boltzmann exact solution are presented in Table 1. As can be seen from the Table, the greatest parameter value variation occurred for the parameter A.

Table 1. INVERSE PROBLEM SOLUTIONS

Run	X_1 (A)	X_2 (A_1)	X_3 (x_0)	X_4 (dx)	F(**X**)
1	15.37	2092.6	0.4096	0.2403	5.10
2	27.82	2089.0	0.4119	0.2382	1.42
3	20.83	2089.9	0.4105	0.2398	1.80
4	30.81	2081.3	0.4110	0.2372	1.56
5	27.16	2089.7	0.4122	0.2386	1.21
6	37.89	2080.4	0.4123	0.2355	2.28
7	29.40	2090.2	0.4125	0.2383	3.10
8	26.51	2092.2	0.4126	0.2392	1.70
9	34.49	2082.3	0.4122	0.2363	1.99
10	26.94	2090.9	0.4125	0.2390	1.54
BOLTZ.	**23.00**	**2087.0**	**0.4100**	**0.2380**	**0.00**

Figure 7 shows the 10 solutions retrieved by the GEO + ES hybrid, in terms of the resulting film cooling temperature profiles, identified as C01 to C10. For each one, the graph gives also the respective F(**X**) final value. The original temperature profile is also presented, identified as "BOLTZMANN". As can be seen, all the solutions match very well the original profile, up to a point that they can't be seen isolatedly, meaning that the optimization algorithm was able to solve the inverse optimization problem very well for all the ten runs. It indicates that the algorithm is not sensitive to the random starting point used to start the search. This is also confirmed by the numerical values obtained for F(**X**). Even the worst one, that happened to be C01, had F(**X**)=5.10ºC. Considering that F(**X**) is, in fact, a difference vector composed of 41 elements, and if it is assumed that the differences have homogeneous distribution among all the 41 elements, then, it would mean a discrepancy of just $5.10/\sqrt{41} \cong 0.8$ºC per element (sensor) for the worst match and only $1.21/\sqrt{41} \cong 0.19$ºC per element (sensor) for the best match.

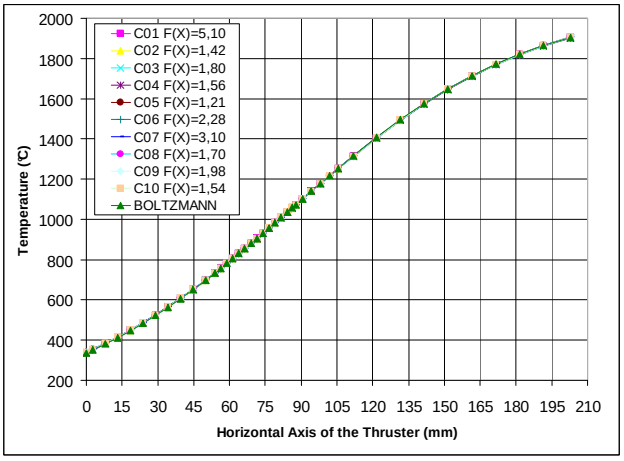

Figure 7. ORIGINAL AND OPTIMIZED FILM COOLING TEMPERATURES.

Figure 8, by its turn, shows the complete profile of temperatures. For the external wall and for the film cooling two temperature profiles are presented. One given by the Boltzmann original parameters (exact solution) and other given by the best solution found obtained by GEO + ES (run number 5). As can be observed in Fig. 8, the two temperature profiles coming from the optimization problem solution fit quite well the corresponding original temperature profiles.

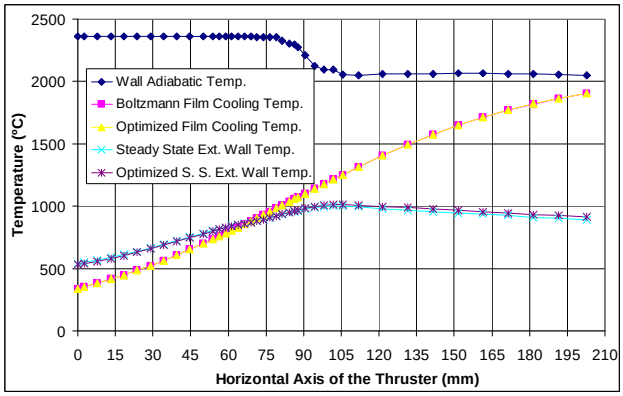

Figure 8. ORIGINAL AND OPTIMIZED TEMPERATURE PROFILES.

7. CONCLUSIONS

In this paper, a new hybrid algorithm called GEO + ES was presented and applied to a real world application. The algorithm was developed in a way to congregate the good convergence properties of a real-valued version of the GEO algorithm with the self-tuning characteristics of the ES methods. A new mutation operator that can be used by any other EA was also introduced and used. The performance of the new algorithm was tested by using it to solve the inverse optimization problem of the estimation of the temperature distribution of the film cooling near the internal wall of a thruster. The results obtained have indicated that GEO + ES was able to solve the problem very well and in a consistent way, independently of the starting point randomly used by the algorithm to start the search. In this way, GEO + ES becomes an important tool for improving the accuracy of the thermal model of the thruster and of all the subsequent thermal simulations to be performed.

ACKNOWLEDGMENTS

The authors acknowledge the financial support provided by CNPQ – Conselho Nacional de Desenvolvimento Científico e Tecnológico, and FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo.

REFERENCES

[1] Galski, R. L. 2006 Development of Improved, Hybrid, Parallel and Multiobjective Versions of the Generalized Extremal Optimization Method and its Application to the Design of Spatial Systems. Doctoral Thesis (In Portuguese). (INPE-14795-TDI/1238) National Institute for Space Research, São José dos Campos, SP, Brazil.

[2] Koslov, A. A, Hinckel, J. N. 1998 Internal Course Notes at INPE about Thruster Project in Brazil-Russian Cooperation.

[3] Goldberg, D. E. 1989 Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company.

[4] Davis, L. D., De Jong, K., Vose, M. D., and Whitley, L. D. (Editors) 1999 Evolutionary algorithms. The IMA Volumes in Mathematics and its Applications, Vol. 111, Springer-Verlag.

[5] Kirkpatrick, S., Gellat, C. D., and Vecchi, M. P. 1983 Optimization by Simulated Annealing. Science 220 (Number 4598), 671-680.

[6] Freeman, J. A., and Skapura, I. 1991 Neural networks: algorithms, applications and programming techniques. Addison-Wesley, New York.

[7] Castro, L. N. and Timmis, J. 2002 An Artificial Immune Network for Multimodal Function Optimization. Proceedings of the IEEE Congress on Evolutionary Computation (Hawaii, May, 2002). CEC'02. Vol. 1, 699-674.

[8] Bonabeau, E. Dorigoú, M., and Theraulaz, G. 2000 Inspiration for Optimization From Social Insect Behaviour. Nature 406, 39-42.

[9] De Sousa, F. L., and Ramos, F. M. 2002 Function Optimization Using Extremal Dynamics. Proceedings of the 4th International Conference on Inverse Problems in Engineering (cd-rom), Angra dos Reis, RJ, Brazil.

[10] De Sousa, F. L., Ramos, F. M., Paglione, P., and Girardi, R. M. 2003 New stochastic algorithm for design optimization. AIAA Journal 41 (Number 9), 1808-1818.

[11] De Sousa, F. L., Ramos, F. M., Galski, R. L., and Muraoka, I. 2004 Generalized Extremal Optimization: A New Meta-heuristic Inspired by a Model of Natural Evolution. In Recent Developments in Biologically Inspired Computing, De Castro, L. N. & Von Zuben, F. J. (editors), Idea Group Inc., Hershey, PA, USA, 41-60.

[12] Boettcher, S. and Percus, A. G. 2001 Optimization with Extremal Dynamics. Physical Review Letters 86, 5211-5214.

[13] De Sousa, F. L., Vlassov, V., and Ramos, F. M. 2002 Heat pipe design through generalized extremal optimization. Proceedings of the IX Brazilian Congress of Engineering and Thermal Sciences (Caxambu, MG, Brazil, 2002). ENCIT 2002.

[14] Muraoka I., Galski R.L., de Sousa F.L., and Ramos F.M. 2006 Stochastic Spacecraft Thermal Design With Low Computational Cost. Journal of Spacecraft and Rockets 43 (n. 6), 1248-1257. (ISSN 1533-6794).

[15] Vlassov V.V., de Sousa F.L., and Takahashi W.K. 2006 Comprehensive Optimization of a Heat Pipe Radiator Assembly Filled with Ammonia or Acetone. International Journal of Heat and Mass Transfer 49, 4584-4595.

[16] De Sousa F.L., Muraoka I., and Galski R.L. 2007 On The Optimal Positioning of Electronic Equipment In Space Platforms. Proceedings of the 19th International Congress of Mechanical Engineering (CDROM), COBEM 2007, 5-9 November, Brasília, Brazil.

[17] Mainenti I., Souza L.C.G., de Sousa F.L., Kuga H.K., Galski R.L. 2007 Satellite Attitude Control Using The Generalized Extremal Optimization With a Multi-Objective Approach. Proccedings of the 19th International Congress of Mechanical Engineering (CDROM), COBEM 2007, 5-9 November, Brasília, Brazil.

[18] Mossi A.C., Schneider P.S., Francis H.R.F., de Sousa F.L., Silva Neto A.J. 2007 Application of The Generalized Extremal Optimization (GEO) Algorithm in a Illumination Inverse Design. Proceedings of the 19th International Congress of Mechanical Engineering, COBEM 2007, 5-9 November, Brasília, Brazil.

[19] Abreu B.T., Martins E., de Sousa F.L. 2007 Generalized Extremal Optimization: an attractive alternative for test data generation. Proceedings of the Genetic and Evolutionary Computation Conference 2007 (Accepted as a Poster), p. 1138-1138, ISBN:978-1-59593-697-4. London, England.

[20] Bak, P. and Sneppen, K. 1993 Punctuated equilibrium and criticality in a simple model of evolution. Physical Review Letters 71 (n. 24), 4083-4086.

[21] Bak, P. 1996. How nature works. Copernicus, Springer-Verlag, New York.

[22] Eiben, A. E. and Smith, J. E. 2003 Introduction to evolutionary computing. Springer-Verlag, Berlin, Germany.

[23] Potter, M.; De Jong, K.A., A Cooperative Coevolutionary Approach to Function Optimization, Proceedings of the Third Parallel Problem Solving From Nature, Springer-Verlag, pp. 249-257, 1994.

[24] Someya H.; Yamamura, M. Genetic Algorithm with search area adaptation for the function optimization and its experimental analysis. Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, South Korea, v.2, pp. 933-940, 2001.

[25] Ballester, P. J.; Carter, J. N. An effective real-parameter Genetic Algorithm with parent centric normal crossover for multimodal optimization. Proceedings of GECCO 2004, Seattle, LNCS 3102, 2004, pp. 901-913.

[26] Patire Júnior, H., Galski R.L., De Sousa F.L., Hinckel, J.N., Lacava, P.T., and Ramos, F.M. 2008 Film Cooling Temperature Estimation of a 200n Hydrazine Thruster by an Inverse Problem Approach. Proceedings of the 12th Brazilian Congress of Thermal Sciences and Engineering, ENCIT 2008, 10-14 November, Belo Horizonte, MG, Brazil.

[27] Dieter K. H., David H. H. 1992 Modern Engineering for Design of Liquid Propellant Rockets Engines, AIAA, pg 84.

[28] Gilmore, D.G. 1994. Fundamentals of Thermal Modeling. In Satellite Thermal Control Handbook. The Aerospace Corporation Press, El Segundo, CA, 5/21-5/39.

[29] Cardoso, H.P., Muraoka, I., Bastos, J.L.F., Bambace, L.A.W., Oliveira Filho, O.B., and Leite, R.M.G. 1990. PCTER Thermal Analysis Software. User's Manual (In Portuguese), INPE, São José dos Campos, SP, Braz