PREPUBLICACIONES DEL DEPARTAMENTO
DE MATEMÁTICA APLICADA
UNIVERSIDAD COMPLUTENSE DE MADRID
MA-UCM 2008-06

OPTIMIZING INITIAL GUESSES TO IMPROVE GLOBAL MINIMIZATION

B. Ivorra, B. Mohammadi , A. M. Ramos

Febrero -2008

http://www.mat.ucm.es/deptos/ma
e-mail:matemática_aplicada@mat.ucm.es

Optimizing Initial Guesses to Improve Global Minimization

Benjamin Ivorra [†], Bijan Mohammadi [‡], Angel Manuel Ramos[†]

[†] Departamento de Matemática Aplicada,
Universidad Complutense de Madrid,
3 Plaza de Ciencias, 28040, Madrid, Spain
[‡]Institut de Mathématiques et de Modélisation de Montpellier
Université de Montpellier II, 34095, Montpellier, France

ABSTRACT- In this paper, we envision global optimization as finding, for a given calculation complexity, a suitable initial guess of a considered optimization algorithm. One can imagine that this possibility clearly improve the capacity of existing optimization algorithms, including stochastic ones. This approach is validated on several large dimension nonlinear minimization problems. Results are compared with those obtained by a genetic algorithm.

KEYWORDS- Global optimization, Dynamical Systems, Semi-Deterministic Algorithms, Genetic Algorithms.

1 Introduction

Many optimization algorithms can be viewed as discrete forms of Cauchy problems for a system of Ordinary Differential Equations (**ODE**) in the space of control parameters [1, 2]. In situations where several local minima exist, those algorithms perform better if the initial condition belongs to the attraction basin of the infimum. This shows the importance of the initial guesses in optimization algorithms. Hence, we would like to see if it is possible to provide a suitable initial condition in order to improve the efficiency of existing optimization methods.

In this work, we discuss different ways to implement this idea for three classes of existing deterministic and nondeterministic minimization algorithms ([1, 3]). In particular, we focus on the hybridization with Genetic Algorithms (**GA**). Indeed, GAs have received tremendous interest in recent years [4, 5, 6], but their high computational complexity and slow convergence are still drawbacks. One would like therefore to enhance their performance by providing better initial populations.

2 Global optimization method

We introduce the following minimization problem:

$$\min_{x\in\Omega} h_0(x) \tag{1}$$

where $h_0 : \Omega \to \mathbb{R}$ is the cost function and x is the optimization parameter belonging to an admissible space $\Omega \subset \mathbb{R}^N$, with $N \in \mathbb{N}$. We assume $h_0 \in C^0(\Omega, \mathbb{R})$ is a coercive function (i.e. $\lim_{\|x\|\to+\infty} h_0(x) = +\infty$).

We consider an optimization algorithm $A_0 : V \to \Omega$, called 'core optimization algorithm', to solve (1). Here V is the space where we can choose the initial condition for A_0 (various examples are given in Section 2.3). The optimization parameters of A_0 (such as the stopping criteria parameters, etc...) are chosen by the user at the beginning.

We assume the existence of a suitable initial condition $v \in V$ such that the output returned by $A_0(v)$ approaches a solution of (1). In this case, solving numerically (1) with the considered core optimization algorithm means to solve

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ A_0(v) \in \text{argmin}_{x\in\Omega} h_0(x). \end{cases} \tag{2}$$

In order to solve (2), we propose to use a multi-layer semi-deterministic algorithm based on line search methods (see, for instance, [1]) called, for the sake of simplicity, 'Semi-Deterministic Algorithm' (**SDA**).

2.1 General description of the SDA

We introduce $h_1 : V \to \mathbb{R}$ by

$$h_1(v) = h_0(A_0(v)). \tag{3}$$

Thus problem (2) can be rewritten as:

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ v \in \text{argmin}_{w\in V} h_1(w). \end{cases} \tag{4}$$

A geometrical representation of $h_1(.)$ in one dimension is shown in Figure 1 for a situation where the core optimization algorithm is the steepest descent algorithm applied with a large number of iterations. We see that $h_1(.)$ is discontinuous with plateaus. Indeed, the same point is reached by the algorithm starting from any of the points of the same attraction basin. Furthermore, $h_1(.)$ is discontinuous where the functional reaches a local maximum. One way to minimize such a kind of function, in the one dimensional case, is to consider line search optimization methods (such as secant method or dichotomy [1]).

Thus, in order to solve (4), we introduce the algorithm $A_1 : V \to V$ that, for each $v_1 \in V$ return $A_1(v_1)$ given by

Step 1- Choose v_2 randomly in V.

Step 2- Find $v \in \text{argmin}_{w \in \mathcal{O}(v_1,v_2)} h_1(w)$, where $\mathcal{O}(v_1, v_2) = \{v_1 + t(v_2 - v_1), t \in \mathbb{R}\} \cap V$, using a line search method.

Step 3- Return v.

The line search minimization algorithm in A_1 and its corresponding parameters are defined by the user.

In fact, we are interested to perform a multi-directional search of the solution of (2). To do so, we add a layer external to the algorithm A_1 by considering the following methodology:

We define $h_2 : V \to \mathbb{R}$ by

$$h_2(v) = h_1(A_1(v)) \tag{5}$$

Then we consider the problem:

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ v \in \text{argmin}_{w \in V} h_2(w). \end{cases} \tag{6}$$

To solve (6) we use the two-layer algorithm $A_2 : V \to V$ that, for each $v_1 \in V$ return $A_2(v_1)$ given by

Step 1- Choose v_2 randomly in V.

Step 2- Find $v \in \text{argmin}_{w \in \mathcal{O}(v_1,v_2)} h_2(w)$ using a line search method.

Step 3- Return v.

As previously, the line search minimization algorithm in A_2 and its corresponding parameters are defined by the user. Due to the fact that the line search direction $\mathcal{O}(v_1, v_2)$ in A_1 is constructed randomly, the algorithm A_2 perform a multi-directional search of the solution of (2).

This construction can be pursued recursively by defining for $i = 3, 4, ...$

$$h_i(v) = h_{i-1}(A_{i-1}(v)) \tag{7}$$

and considering the problem:

$$\begin{cases} \text{Find } v \in V \text{ such that} \\ v \in \text{argmin}_{w \in V} h_i(w). \end{cases} \tag{8}$$

Problem (8) is solved using the i-layer algorithm $A_i : V \to V$ that, for each $v_1 \in V$ return $A_i(v_1)$ given by

Step 1- Choose v_2 randomly in V.

Step 2- Find $v \in \text{argmin}_{w \in \mathcal{O}(v_1,v_2)} h_i(w)$ using a line search method.

Step 3- Return v.

As before, line search method used in Step 2 and its corresponding parameters are defined by the user.

In practice we run A_i with suitable stopping criteria and with $v_1 \in V$ arbitrary (or $v_1 \in V$ a good initial guess, if available).

The choice of the random technique used to generate v_2 during Step 1 of A_i is important and could depend on the shape of h_0. For instance, if we know that h_0 has several local minima in Ω with small attraction basins, it seems appropriate to generate v_2 in a small neighborhood of v_1.

The line search minimization algorithm used during Step 2 of A_i should depend on the properties of h_0. In the sequel, we present an implementation of the SDA, considering various core optimization algorithms, in the particular case where h_0 is a non negative function with zero as the minimum value (which often occurs in industrial problems [7, 8, 9, 10]).

2.2 SDA implementation with 1st order descent core optimization algorithms

We consider core optimization algorithms A_0 that come from the discretization of the following initial value problem [2, 11, 1]:

$$\begin{cases} M(x(t),t)x_t(t) = -d(x(t)), \quad t \geq 0, \\ x(0) = x_0, \end{cases} \tag{9}$$

where t is a fictitious time, $x_t = \frac{\mathrm{d}x}{\mathrm{d}t}$, $M : \Omega \times \mathbb{R} \to M_{N\times N}$ (where $M_{N\times N}$ denotes the set of matrix $N \times N$) and $d : \Omega \to \mathbb{R}^N$ is a function giving a descent direction. For example, assuming $h_0 \in C^1(\Omega, \mathbb{R})$, if $d = \nabla h_0$ and $M(x,t) = Id$ (the identity operator) for all $(x,t) \in \Omega \times \mathbb{R}$ we recover the steepest descent method.

According to previous notations, we use $V = \Omega$ and denote by $A_0(x_0) := A_0(x_0; t_0, \epsilon)$ the solution returned by the core optimization algorithm starting from the initial point $x_0 \in \Omega$ after $t_0 \in \mathbb{N}$ iterations and considering a stopping criterion defined by $\epsilon \in \mathbb{R}$.

We point out that for this choice of A_0, problem (2) is trivially admissible since, for instance, any $x_0 \in \mathrm{argmin}_{x\in\Omega} h_0(x)$ is a solution. In Section 2.3, we show a different case where the solvability of the corresponding problem is not trivial.

We consider a particular implementation of the algorithms A_i, $i = 1, 2, ...$, introduced previously. For $i = 1, 2, ...$, $A_i(v_1)$ is implemented with a secant method (a low-cost method well adapted to find the zeros of a function [1]) in order to perform the line search. It reads:

Step 1- Choose $v_2 \in \Omega$ randomly.

Step 2- For l from 1 to $t_{l_i} \in \mathbb{N}$ (large enough) execute:

Step 2.1- If $h_i(v_l) = h_i(v_{l+1})$ go to **Step 3**

Step 2.2- Set v_{l+2} =proj$_\Omega(v_{l+1} - h_i(v_{l+1})\frac{v_{l+1}-v_l}{h_i(v_{l+1})-h_i(v_l)})$
where proj$_\Omega : \mathbb{R} \to \Omega$ is a projection algorithm over Ω defined by the user.

Step 3- Return the output: argmin$\{h_i(v_m), m = 1, ..., t_{l_i}\}$

A geometrical representation of one execution of the algorithm A_1 in one dimension is shown in Figure 1.

Remark. When the minimum of h_0 is unknown we can change Step 2.2 by other methods as the steepest descent iteration starting from v_{l+1} and using $-\frac{v_{l+1}-v_l}{h_i(v_{l+1})-h_i(v_l)}$ as the descent direction.

2.3 SDA implementation with 2nd order descent core optimization algorithms

In order to keep an exploratory character during the optimization process, allowing to escape from attraction basins, we could use variants of previous core optimization methods after adding second order derivatives to the initial value problem (9). More precisely, we consider core optimization algorithms A_0 coming from the discretization of:

$$\begin{cases} \eta x_{tt}(t) + M(x(t), t)x_t(t) = -d(x(t)), \quad t \geq 0, \\ \quad x(0) = x_0, \quad x_t(0) = x_{t,0}, \end{cases} \tag{10}$$

with $x_{tt} = \frac{\mathrm{d}^2 x}{\mathrm{d}t^2}$ and $\eta \in \mathbb{R}$. For instance, assuming $h_0 \in C^2(\Omega, \mathbb{R})$, when $d = \nabla h_0$ and $M(x, t) = Id$ for all $(x, t) \in \Omega \times \mathbb{R}$ we recover the '*heavy ball*' dynamical system [2, 12].

Here, $A_0(x_0, x_{t,0}) := A_0(x_0, x_{t,0}; t_0, \epsilon)$ denotes the solution returned by the core optimization algorithm starting from the initial point $x_0 \in \Omega$ with an initial velocity of $x_{t,0} \in \mathbb{R}^N$, after $t_0 \in \mathbb{N}$ iterations and considering a stopping criterion given by $\epsilon \in \mathbb{R}$.

In this case, we propose two different formulations for (2). The first one is given by

$$\begin{cases} \text{Find } x_0 \in \Omega \text{ such that} \\ A_0(x_0, x_{t,0}) \in \text{argmin}_{w\in\Omega} h_0(w), \end{cases} \tag{11}$$

where $x_{t,0}$ is fixed. The second one is given by

$$\begin{cases} \text{Find } x_{t,0} \in \mathbb{R}^N \text{ such that} \\ A_0(x_0, x_{t,0}) \in \text{argmin}_{w\in\Omega} h_0(w), \end{cases} \tag{12}$$

where x_0 is fixed.

The existence of $x_0 \in \Omega$ such that (11) admits a solution is trivial (as in Section 2.2). In the second case, under convenient hypotheses, it can be proved the existence of $x_{t,0} \in \mathbb{R}^N$ such that (12) admits a numerical solution, as stated in the following Theorem:

Theorem 1 *Let $h_0 : \mathbb{R}^N \to \mathbb{R}$ be a C^2-function having a minimum, which is reached at $x_m \in \mathbb{R}^N$. Then, for every $(x_0, \epsilon) \in \mathbb{R}^N \times \mathbb{R}^+$, there exists $(\sigma, \tau_b) \in \mathbb{R}^N \times \mathbb{R}^+$ such that the solution of the following dynamical system:*

$$\begin{cases} \eta x_{tt}(t) + x_t(t) = -\nabla h_0(x(t)), \quad t \geq 0, \\ x(0) = x_0, \\ x_t(0) = \sigma, \end{cases} \tag{13}$$

with $\eta \in \mathbb{R}$, passes at time τ_b into the ball $B_\epsilon(x_m)$.

Proof :

We assume $x_0 \neq x_m$ ($x_0 = x_m$ is a trivial case). Let $\delta \geq 0$, we consider the initial value problem:

$$\begin{cases} \eta y_{\delta,tt}(t) + \delta y_{\delta,t}(t) = -\delta^2 \nabla h_0(y_\delta(t)), \quad t \geq 0, \\ y_\delta(0) = x_0, \\ y_{\delta,t}(0) = \varrho(x_m - x_0), \end{cases} \tag{14}$$

with $\varrho \in \mathbb{R}^+ \backslash \{0\}$. Let us show that y_δ passes at some time into the ball $B_\epsilon(x_m)$:

- If $\delta = 0$, we obtain the following system :

$$\begin{cases} \eta y_{0,tt}(t) = 0, \quad t \geq 0, \\ y_0(0) = x_0, \\ y_{t,0}(0) = \varrho(x_m - x_0). \end{cases} \tag{15}$$

  System (15) describes a straight line of origin x_0 and passing at some time $\tau_\varrho \in \mathbb{R}^+$ by the point x_m, i.e. $y_0(\tau_\varrho) = x_m$.

- If $\delta \neq 0$, system (14) can be rewritten in the form $w_t(t) = (y_{\delta,t}, -\delta y_{\delta,t}(t) - \delta^2 \nabla h_0(y_\delta(t))) = f(t, w(t), \delta)$, with $w(t) = (y_\delta(t), \eta y_{\delta,t}(t))$ and f continuous in t and in δ and Lipschitz continuous in $w(t)$ [12]. Then, applying the Cauchy-Lipschitz theorem (see, for instance, [13]):

$$\lim_{\delta \to 0} |y_\delta(\tau_\varrho) - y_0(\tau_\varrho)| = 0.$$

Thus for every $\epsilon \in \mathbb{R}^+ \backslash \{0\}$, there exists $\tau_\epsilon \in \mathbb{R}^+$ such that for every $\delta \leq \tau_\epsilon$:

$$|y_\delta(\tau_\varrho) - x_m| < \epsilon. \tag{16}$$

Let $\epsilon \in \mathbb{R}^+ \backslash \{0\}$. We consider the change of variable given by $s = \tau_\epsilon t$ and $x(s) = y_{\tau_\epsilon}(\frac{s}{\tau_\epsilon})$. Then system (14) becomes:

$$\begin{cases} \eta x_{ss}(s) + x_s(s) = -\nabla h_0(x(s)), \quad s \geq 0, \\ x(0) = x_0, \\ \dot{x}(0) = \frac{\varrho}{\tau_\epsilon}(x_m - x_0). \end{cases} \tag{17}$$

Let $\tau_b = \tau_\epsilon \tau_\varrho$. Under this assumption, $x(\tau_b) = y_{\tau_\epsilon}(\tau_\varrho)$. Thus, due to (16): $|x(\tau_b) - x_m| < \epsilon$. We have found $\sigma = \frac{\varrho}{\tau_\epsilon}(x_m - x_0) \in \mathbb{R}^N$ and $\tau_b \in \mathbb{R}^+$ such that the solution of system (13) passes at time t_b into the ball $B_\epsilon(x_m)$.

□

In order to determine a solution of (11) or (12), we can consider, for example, the same implementation of algorithms A_i (with $i = 1, 2, 3, ...$) as introduced in Section 2.2.

2.4 SDA implementation with genetic core optimization algorithms

Now, we are interested to study SDA with GA as core optimization algorithm. GAs can be seen as iterations of a coupled system of discrete stochastic ODEs starting from a first family $X^0 = \{x_j^0 \in \Omega, j = 1, ..., N_p\}$ of N_p possible solutions of the optimization problem (see Appendix for a complete description of the algorithm and notations).

Problem (2) can be rewritten as:

$$\begin{cases} \text{Find } X^0 \in V = \Omega^{N_p} \text{ such that} \\ A_0(X^0) \in \text{argmin}_{w \in \Omega} h_0(w) \end{cases} \tag{18}$$

where $A_0(X^0) := A_0(X^0; N_p, N_g, p_m, p_c)$ with N_p, N_g, p_m, p_c parameters of the GA algorithm that here are considered fixed.

The solution of (18) may be determined, for instance, by using the SDA implementation presented in Section 2.2. However, a first numerical study (see [14] for more details) shows that the following variation of previous algorithms A_i (with $i = 1, 2, 3, ...$) is better adapted to the GA case. Let $X_1^0 = \{x_{1,j}^0 \in \Omega, j = 1, ..., N_p\}$. Then $A_i(X_1^0)$ reads:

Step 1- For l going from 1 to $t_{l_i} \in \mathbb{N}$ (large enough) execute:

Step 1.1- Set $o_l = \text{argmin}\{h_0(x) : x \in A_{i-1}(X_l^0)\}$

Step 1.2- We construct $X_{l+1}^0 = \{x_{l+1,j}^0 \in \Omega, j = 1, ..., N_\text{p}\}$ as following:
$\forall j \in \{1, ..., N_\text{p}\}$, if $h_0(o_l) = h_0(x_{l,j}^0)$ set $x_{l+1,j}^0 = x_{l,j}^0$
else set $x_{l+1,j}^0 = \text{proj}_\Omega(x_{l,j}^0 - h_0(o_l)\frac{o_l - x_{l,j}^0}{h_0(o_l) - h_0(x_{l,j}^0)})$
where $\text{proj}_\Omega : \mathbb{R} \to \Omega$ is a projection algorithm over Ω defined by the user.

Step 2- Return the output: $\text{argmin}\{h_i(X_m^0), m = 1, ..., t_{l_i}\}$

This version of the algorithm intends to optimize, individual by individual, the initial population of A_{i-1}. For each individual in the initial population:

- If there is a significant evolution of the cost function value between this individual and the best element found by A_{i-1}, the secant method used in Step 1.2 generates, in the optimized initial population, a new individual closer to this best element.
- If not, the secant method allows to create a new individual far from the current solution given by A_{i-1}.

In Section 3, numerical experiments show that this coupling reduces the computational complexity of GAs. In particular, this permits to consider smaller N_p and N_g than with GAs alone.

3 Numerical examples

In this section, we focus on several benchmark optimization problems (in part presented in [15]) to be solved using SDAs and GAs.

3.1 Parameters in algorithms

We consider various versions of the SDA in the cases when the core optimization algorithm is, respectively, a steepest descent algorithm, a heavy ball algorithm and a genetic algorithm:

- When the steepest descent algorithm is the core optimization algorithm we use the SDA implementation presented in Section 2.2. We consider the cases when the number of layers is $i = 1$ (the algorithm is then denoted by **SDDA-1L** as Semi-Deterministic Descent Algorithm-1 Layer), $i = 2$ (**SDDA-2L**) and $i = 3$ (**SDDA-3L**). We set $t_0 = 10$, $t_{l_1} = 5$, $t_{l_2} = 5$. The number of iterations for the highest layer, t_{l_i}, is set large enough.
- When the heavy ball method is used as the core optimization algorithm, we use the two-layer algorithm A_2, described in Sections 2.2 and 2.3, with $\eta = 0.1$, $t_0 = 10$, $t_{l_1} = 5$ and t_{l_2} large enough. The velocity $x_{t,0}$ is the initial condition to be optimized. This algorithm is denoted by **HBSDA** (Heavy-Ball Semi-Deterministic Algorithm).
- With GA as the core optimization algorithm, we use the algorithm A_2 introduced in Section 2.4 with $t_{l_1} = 5$ and t_{l_2} large enough. This algorithm is denoted by **GSDA** as Genetic Semi-Deterministic Algorithm. In addition, the GA parameters are set as following:
  - The population size is set to $N_p = 10$.
  - The selection is a roulette wheel type [3] proportional to the rank of the individuals in the population.

– The crossover is barycentric in each coordinate with a probability of $p_c = 0.45$.
– The mutation process is non-uniform with a probability of $p_m = 0.35$.
– A one-elitism principle, that consists in keeping the current best individual in the next generation, has also been imposed.

The performances of the previous SDAs are compared to those of two classical GAs. Both GAs use the same stochastic processes than GSDA but have two different sets of parameters:

- First GA denoted by **GA-S1** applied with: $N_p = 180$, $p_c = 0.45$, $p_m = 0.15$.
- Second GA denoted by **GA-S2** applied with: $N_p = 50$, $p_c = 0.5$, $p_m = 0.3$.

3.2 Results

Due to the stochastic aspect of the approach results are average values over 10 simulations with a standard deviation of $\pm 5\%$.

Flat Iso-contour Convex Function (FICF)

$$h_0(x) = \sum_{j=1}^{N} x_j^{2j}, \tag{19}$$

with $x = (x_1, ..., x_N) \in [-5, 5]^N$ and N is successively set to $10, 100$ and 1000. This function is interesting as it is convex with flat iso-contours resulting in a slow convergence of various optimization algorithms (such as the steepest descent). Its minimum equals to 0 and is reached at the origin.

Results are reported in Table 1. As we can observe, combining steepest descent with our technique allows to perform a satisfactory minimization. The choice of a single-layer structure is less time consuming than multi-layer structures.

This benchmark test also points out the difficulty of HBSDA to find accurate results. This is due to the perturbation created by the 2^{nd} order derivative term.

GSDA and GAs are only compared in the cases when $n = 10$ and $n = 100$ as a good convergence is difficult to obtain with genetic based algorithms. GSDA is twice faster than the other genetic methods.

OPT. METH.	SDDA-1L	SDDA-2L	SDDA-3L	HBSDA
N=10 / $rd = 10^{-6}$	100	400	500	600
N=100 / $rd = 10^{-7}$	200	500	600	800
N=1000 / $rd = 10^{-8}$	700	600	700	1600
OPT. METH.	GSDA	GA-S1	GA-S2	
N=10 / $rd = 10^{-5}$	200	500	400	
N=100 / $rd = 10^{-6}$	300	600	600	

Table 1: FICF results: Number of cost function evaluations needed by optimization methods (OPT. METH.) to obtain a reduction by a factor rd of the initial value of the cost function considered in dimension N.

Generalized Rastringin Function (GRF)

$$h_0(x) = \sum_{j=1}^{N} (x_j^2 - \cos(18x_j)), \tag{20}$$

with $x = (x_1, ..., x_N) \in [-5, 5]^N$ and N is successively set to $2, 10$ and 100. This function is a perturbed version of a convex function with a large number of local minima. Its minimum equals to 0 and is reached at the origin.

We can see in Table 2 that performing a multi-directional search of the initial condition provides a faster optimization process. Furthermore, the steepest descent based methods seems to be better adapted to this kind of function.

As previously, GSDA is faster than the classical genetic based methods.

OPT. METH.	SDDA-1L	SDDA-2L	SDDA-3L	HBSDA
N=2 / $rd = 10^{-5}$	1000	1000	900	1000
N=10 / $rd = 10^{-6}$	1500	1400	1000	1500
N=100 / $rd = 10^{-7}$	2000	1800	1200	3000
OPT. METH.	GSDA	GA-S1	GA-S2	
N=2 / $rd = 10^{-4}$	1000	1800	3000	

Table 2: GRF results: Number of cost function evaluations needed by optimization methods (OPT. METH.) to obtain a reduction by a factor rd of the initial value of the cost function considered in dimension N.

Modified Rastringin Function (MRF)

$$h_0(x) = \sum_{j=1}^{2} (\sin(x_j)^2 - \cos(18x_j)), \tag{21}$$

with $x = (x_1, x_2) \in [-3, 3]^2$. The minimum of h_0 is equal to 0 and it is reached at the origin. In this case, we have modified the previous Rastringin function in order to create a perturbed version of a non convex function.

As we can observe in Table 3, using a second order descent method as core optimization algorithm seems to be more adapted to this type of functions. The GSDA still gives better results than the classical GAs.

OPT. METH.	SDDA-1L	SDDA-2L	SDDA-3L	HBSDA
$rd = 10^{-5}$	Fail	1000	1000	600
OPT. METH.	GSDA	GA-S1	GA-S2	
$rd = 10^{-4}$	700	800	1000	

Table 3: MRF results: Number of cost function evaluations needed by optimization methods (OPT. METH.) to obtain a reduction by a factor rd of the initial value of the cost function.

Modified Rosenbrock Function (MROF)

$$h_0(x) = 40 + 100(x_2 - x_1^2)^2 + (1 - x_1)^2 - 400 \exp^{-10((x_1+1)^2+(x_2+1)^2)}, \quad (22)$$

with $x = (x_1, x_2) \in [-2, 2]^2$. This function is compound by a large attraction basin of a local minimum and a small attraction basin of the global minimum. Its minimum is equal to 0 and it is reached at $(1, 1)$.

This problem is well adapted to study the efficiency of stochastic methods. As we can observe in Table 4, GSDA is faster than all other methods and the other SDA techniques also provide interesting alternatives to classical GAs.

OPT. METH.	SDDA-1L	SDDA-2L	SDDA-3L	HBSDA
$rd = 10^{-4}$	Fail	3000	2000	3000
OPT. METH.	GSDA	GA-S1	GA-S2	
$rd = 10^{-4}$	1000	5000	2500	

Table 4: MROF results: Number of cost function evaluationsneeded by optimization methods (OPT. METH.) to obtain a reduction by a factor rd of the initial value of the cost function.

3.3 Industrial Applications

In addition to the previous benchmark results, SDDA and GSDA have been validated on several industrial problems involving numerous local minima such as: optical fiber synthesis [10], shape optimization of fast-micro-fluidic

mixer devices [9], temperature and pollution control in a bunsen flame [16], portfolio risk management [8], control problem of the Burgers equation [7], shape optimization of coastal structures [17], shape optimization under aerodynamic and acoustic constraints for internal and external flows [11]. For all those problems, those algorithms have provided an affordable approach and satisfactory numerical solutions.

4 Conclusions

A new semi-deterministic global optimization algorithm has been introduced. This algorithm provides suitable initial conditions for existing minimization algorithms. It has been coupled with both deterministic and non-deterministic algorithms. Numerical results show that this coupling upgrade the performance of the considered algorithms.

A Matlab© version of the algorithms presented in this paper are included in the free optimization package "Global Optimization Platform" which can be downloaded at

http://www.mat.ucm.es/momat/software.htm

5 Acknowledgements

This work was carried out with financial support from the "Institut de Math-ématiques et de Modélisation de Montpellier"; the Spanish "Ministry of Education and Science" under the projects No. MTM2007-64540 and "Ingenio Matematica (i-MATH)" No. CSD2006-00032 (Consolider-Ingenio 2010); and the "Dirección General de Universidades e Investigación de la Consejería de Educación de la Comunidad de Madrid" and the "Universidad Complutense de Madrid" in Spain, under the project No. CCG07-UCM/ESP-2787.

Appendix: Genetic algorithms

We briefly recall the general form of GAs and show that they can be seen as iterations of a coupled system of discrete stochastic ODEs.

GAs approximate the global minimum of the cost function h_0 through a stochastic process based on an analogy with the Darwinian evolution of species [3]. A first family, called 'population', $X^0 = \{x_l^0 \in \Omega, j = 1, ..., N_p\}$ of N_p possible solutions of the optimization problem, called 'individuals', is randomly generated in the search space Ω. Starting from this population, we build recursively N_{gen} new populations, called generations, $X^i = \{x_l^i \in \Omega, j = 1, ..., N_p\}$ with $i = 1, .., N_{\text{gen}}$ through three stochastic steps, called

selection, crossover and mutation. More precisely we present here a matrix-form approach for GAs:

We first rewrite X^i using the following (N_p, N)-real valued matrix form:

$$X^i = \begin{bmatrix} x_1^i(1) & \dots & x_1^i(N) \\ \vdots & \ddots & \vdots \\ x_{N_p}^i(1) & \dots & x_{N_p}^i(N) \end{bmatrix} \tag{23}$$

Selection: Each individual, x_l^i is ranked with respect to its cost function value $h_0(x_l^i)$ (i.e. the lower is its value of $h_0(x_l^i)$ the higher is the ranking). Then N_p individuals are randomly selected (individuals with better ranking have higher chances to be selected), with eventual repetitions, to become 'parents'.

Introducing a binary (N_p, N_p)-matrix $\mathcal{S}^i$, generated according to previous ranking and selection processes, with $\mathcal{S}_{j,k}^i = 1$ if the kth individual of X^i is the selected 'parent' number j and $\mathcal{S}_{j,k}^i = 0$ otherwise, we define:

$$X^{i+1/3} = \mathcal{S}^i X^i. \tag{24}$$

Crossover: This process leads to a data exchange between two 'parents' and the apparition of two new individuals called 'children'. We determine, with a probability p_c, if two consecutive parents in $X^{i+1/3}$ should exchange data or if they are directly copied into the intermediate population $X^{i+2/3}$.

Introduce a real-valued (N_p, N_p)-matrix $\mathcal{C}^i$ where for each couple of consecutive lines $(2j-1, 2j)$ $(1 \leq j \leq \frac{N_p}{2}$ in case N_p is even or $1 \leq j \leq \frac{N_p-1}{2}$ in case N_p is odd), the coefficients of the $(2j-1)$th and $2j$th rows are given by:

$$\mathcal{C}_{2j-1,2j-1}^i = \lambda_1, \quad \mathcal{C}_{2j-1,2j}^i = 1-\lambda_1, \quad \mathcal{C}_{2j,2j-1}^i = \lambda_2, \quad \mathcal{C}_{2j,2j}^i = 1-\lambda_2$$

In this expression:

- $\lambda_1 = \lambda_2 = 1$ if parents are directly copied (with a probability $1 - p_c$).
- λ_1 and λ_2 are randomly chosen in $]0, 1[$ if a data exchange occurs between the two parents (with probability p_c).

Other coefficients of $\mathcal{C}^i$ are set to 0. If N_p is odd, the N_pth parent is directly copied, i.e $\mathcal{C}_{N_p,N_p}^i = 1$.

This step can be summarized as:

$$X^{i+2/3} = \mathcal{C}^i X^{i+1/3}. \tag{25}$$

Mutation: This process leads to new parameter values for some individuals of the population. More precisely, each child is modified (or mutated) with a fixed probability p_m.

Introduce for instance a random perturbation matrix $\mathcal{E}^i$ with an jth line equal to:

- a random vector $\epsilon_j \in \mathbb{R}^N$, according to the admissible space Ω, if a mutation is applied to the ith child (with probability p_m).
- 0 if no mutation is applied to the jth child (with probability 1-p_m).

This step can then take the following form:

$$X^{i+1} = X^{i+2/3} + \mathcal{E}^i. \tag{26}$$

Therefore, the new population can be written as:

$$X^{i+1} = \mathcal{C}^i \mathcal{S}^i X^i + \mathcal{E}^i. \tag{27}$$

At the end of the algorithm, after N_g iterations, the GA returns an output denoted by $A_0(X^0, N_p, N_g, p_m, p_c, \epsilon) = \text{argmin}\{h_0(x_j^i)/x_j^i \in X^i, i = 1, ..., N_p, j = 1, ..., N_g)$.

With these three basic evolution processes, it is generally observed that the best obtained individual is getting closer after each generation to the optimal solution of the problem [3].

These algorithms do not require sensitivity computation, perform global and multi-objective optimization and are easy to parallelize. However, their drawbacks remain their weak mathematical background, their computational complexity, their slow convergence and their lack of accuracy.

As a fine convergence is difficult to achieve with GA based algorithms, it is recommended when it is possible, to complete the GA iterations by a descent method. This is especially useful when the functional is flat around the infimum [6].

References

[1] B. Mohammadi and J-H. Saiac. *Pratique de la simulation numérique.* Dunod, 2002.

[2] H. Attouch and R. Cominetti. A dynamical approach to convex minimization coupling approximation with the steepest descent method. *Journal of Differential Equations*, 128(2):519–540, 1996.

[3] D. Goldberg. *Genetic algorithms in search, optimization and machine learning.* Addison Wesley, 1989.

[4] M. Ericsson, , M.G.C. Resende, and P.M. Pardalos. A genetic algorithm for the weight setting problem in ospf routing. *J. Comb. Optim.*, 6(3):299–333, 2002.

[5] C. M. Fonseca and J. Fleming. An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.

[6] L. Dumas, V. Herbert, and F. Muyl. Hybrid method for aerodynamic shape optimization in automotive industry. *Computers and Fluids*, 33(5):849–858, 2004.

[7] B. Ivorra, A.M. Ramos, and B. Mohammadi. Semideterministic global optimization method: Application to a control problem of the burgers equation. *Journal of Optimization Theory and Applications*, 135(3):549–561, 2007, DOI: 10.1007/s10957-007-9251-8.

[8] B. Ivorra, B. Mohammadi, and A.M. Ramos. Optimization strategies in credit portfolio management. *Journal Of Global Optimization*, Accepted, in Early view, to be published, DOI: 10.1007/s10898-007-9221-6.

[9] B. Ivorra, B. Mohammadi, D.E. Santiago, and J.G. Hertzog. Semideterministic and genetic algorithms for global optimization of microfluidic protein folding devices. *International Journal of Numerical Method in Engineering*, 66(2):319–333, 2006, DOI: 10.1002/nme.1562.

[10] B. Ivorra, B. Mohammadi, L. Dumas, O. Durand, and P. Redont. Semideterministic vs. genetic algorithms for global optimization of multichannel optical filters. *International Journal of Computational Science for Engineering*, 2(3):170–178, 2006, DOI: 10.1504/IJCSE.2006.012769.

[11] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2001.

[12] H. Attouch, X. Goudou, and P. Redont. The heavy ball with friction method. i: The continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system. *Commun. Contemp. Math.*, 2(1):1–34, 2000.

[13] Verhulst F. *Nonlinear differential equations and dynamical systems.* Springer-Verlag., 1990.

[14] B. Ivorra. Semi-deterministic global optimization. *PhD. University of Montpellier 2*, 2006.

[15] C.A. Floudas, P.M. Pardalos, C.S. Adjiman, W.R. Esposito, Z. Gumus, S.T. Harding, J.L. Klepeis, C.A. Meyer, and C.A. Schweiger. *Handbook of test problems in local and global optimization.* Kluwer Academic Publishers, 2000.

[16] L. Debiane, B. Ivorra, B. Mohammadi, F. Nicoud, A. Ern, T. Poinsot, and H. Pitsch. A low-complexity global optimization algorithm for temperature and pollution control in flames with complex chemistry. *International Journal of Computational Fluid Dynamics*, 20(2):93–98, 2006, DOI: 10.1080/10618560600771758.

[17] D. Isebe, F. Bouchette, P. Azerad, B. Ivorra, and B. Mohammadi. Optimal shape design of coastal structures. *International Journal of Numerical Method in Engineering*, Accepted, in Early view, to be published, DOI: 10.1002/nme.2209.

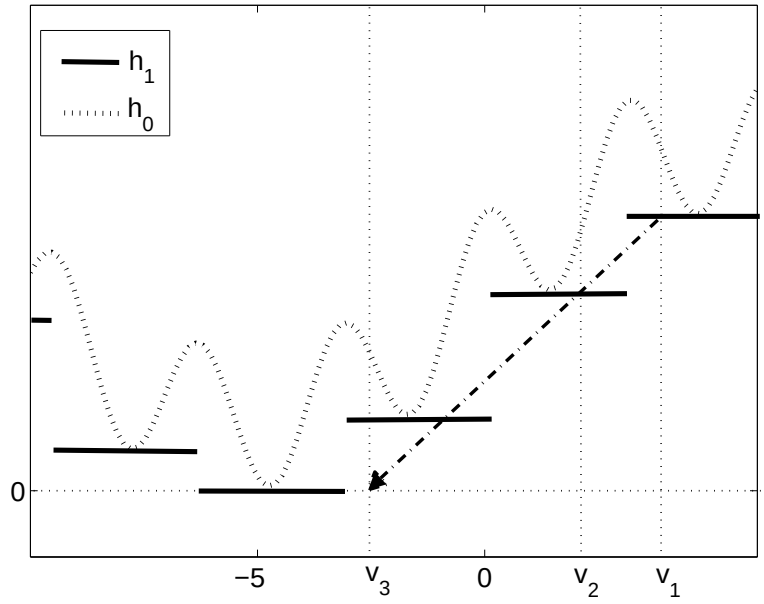

Figure 1: **(....)** $h_0(x) = \frac{1}{2}\cos(2x) + \sin(\frac{1}{3}x) + 1.57$ for $x \in \Omega = V = [-10, 6]$. **(—-)** Geometrical representation of $h_1(.)$ when the steepest descent method is used as core optimization algorithm with a large number of iterations. **(-.-.)** Geometrical representation of one execution of the algorithm $A_1(v_1)$, written in Section 2.2, when v_1 is given and $t_{l_1} = 1$. v_2 is generated randomly $\in [-10, 6]$ during the first Step of the algorithm. v_3 is built by the secant method performed during the Step 2.2. During the Step 3, as $h_1(v_3)$ is lower than $h_1(v_1)$ and $h_1(v_2)$, v_3 is considered as the best initial condition and is returned as the output.

PREPUBLICACIONES DEL DEPARTAMENTO
DE MATEMÁTICA APLICADA
UNIVERSIDAD COMPLUTENSE DE MADRID
MA-UCM 2007

1. DISSIPATIVE SOLITONS THAT CANNOT BE TRAPPED, R. Pardo and V. M. Pérez-García

2. A SEMI-DETERMINISTIC GLOBAL OPTIMIZATION METHOD. APPLICATION TO A CONTROL PROBLEM OF THE BURGERS EQUATION, COMPARING WITH OTHER METHODS, B. Ivorra, A. M. Ramos and B. Mohammadi

3. SEMISTABLE EXTREMAL GROUND STATES FOR NONLINEAR EVOLUTION EQUATIONS IN UNBOUNDED DOMAINS, A. Rodríguez-Bernal and A. Vidal-López

4. BLOW-UP WITH LOGARITHMIC NONLINEARITIES, R. Ferreira, A. de Pablo, and J. Rossi

5. RAPIDLY VARYING BOUNDARIES IN EQUATIONS WITH NONLINEAR BOUNDARY CONDITIONS. THE CASE OF A LISPCHITZ DEFORMATION, J. M. Arrieta and S. M. Bruschi

6. OPTIMIZATION STRATEGIES IN CREDIT PORTFOLIO MANAGEMENT, B. Ivorra, B. Mohammadi and A. M. Ramos

7. MODELLING AND SIMULATION OF HIGH PRESSURE PROCESSES IN FOOD ENGINEERING , S. J.Alvarez, J.A. Infante, B. Ivorra, A.M. Ramos and J.M. Rey

8. SPECTRAL INFORMATION RETRIEVAL FROM INTEGRATED BROADBAND PHOTODIODE UV MEASUREMENTS, L.Vázquez, M.P. Zorzano and S.Jiménez

9. THE ANTI-SYMMETRY PRINCIPLE FOR QUASI-STATIC CRACK PROPAGATION IN MODE III. G. E. Oleaga

10. FRACTIONAL TAYLOR SERIES FOR CAPUTO FRACTIONAL DERIVATIVES. CONSTRUCTION OF NUMERICAL SCHEMES, D. Usero

11. AN ABSOLUTELY CONTINUOS INVARIANT MEASURE FOR A FAMILY OF NONUNIFORMLY EXPANDING MULTIMODAL MAPS, I.Bosch

PREPUBLICACIONES DEL DEPARTAMENTO
DE MATEMÁTICA APLICADA
UNIVERSIDAD COMPLUTENSE DE MADRID
MA-UCM 2008

1. SHAPE OPTIMIZATION OF GEOTEXTILE TUBES FOR SANDY BEACH PROTECTION, D. Isebe, P. Azerad, F. Bouchette, B. Ivorra and B. Mohammadi

2. ON THE FINITE TIME EXTINTION PRHENOMENON FOR SOME NONLINEAR FRACTIONAL EVOLUTION EQUATIONS, J.I. Diaz, T. Pierantozzi and L. Vázquez

3. SEMICLASSICAL MEASURES AND THE SCHRÖDINGER FLOW ON COMPACT MANIFOLDS, F.Macià

4. INVERSE PROBLEMS IN HEAT EXCHANGE PROCESSES, A. Fraguela, J.A. Infante, A.M.Ramos and J.M.Rey

5. EXTREMAL EQUILIBRIA FOR DISSIPATIVE PARABOLIC EQUATIONS IN LOCALLY UNIFORM SPACES, J. Cholewa and A. Rodríguez-Bernal

6. OPTIMIZING INITIAL GUESSES TO IMPROVE GLOBAL MINIMIZATION, B. Ivorra, B. Mohammadi and A. M. Ramos.