

# Semi-deterministic and genetic algorithms for global optimization of microfluidic protein folding devices

Benjamin Ivorra <sup>\*</sup>, David E. Hertzog <sup>†</sup>  
Bijan Mohammadi <sup>\*</sup>, Juan G. Santiago <sup>†</sup>

Draft for IJNME

*Abstract - In this paper we reformulate global optimization problems in terms of boundary value problems (BVP). This allows us to introduce a new class of optimization algorithms. Indeed, current optimization methods, including non-deterministic ones, can be seen as discretizations of initial value problems for differential equations, or systems of differential equations. Furthermore, in order to reduce computational time approximate state and sensitivity evaluations are introduced during optimization. Lastly, we demonstrated the efficacy of two algorithms, included in the former class, on two academic test cases and on the design of a fast microfluidic protein folding device. The aim of the latter design is to reduce mixing times of proteins to microsecond timescales. Results are compared with those obtained with a classical genetic algorithm.*

**Keywords:** Shape optimization, Global optimization, Dynamical systems, Boundary value problem, Microfluidic mixers.

## 1 Introduction

Global solution of minimization problems is of great practical importance and this is one of the reason why evolutionary or genetic algorithms (GA) received tremendous interest in recent years [1, 2]. The main difficulties with these algorithms are their complexity in term of the number of evaluations of the function, their lack of accuracy and their slow convergence.

Minimization algorithms can be shown being discrete forms of Cauchy problems for ordinary differential equation (or system of equations) for control parameters. We will see that if one introduces extra information on the infimum global optimization can be formulated as boundary value problems for the same equations [3, 4, 5]. A motivating idea is therefore to apply algorithms for the solution boundary value problems to global optimization. This is our aim through this paper. In particular, we consider the design of microfluidic protein folding devices. The algorithm issued from our boundary value problem analysis is compared to two genetic algorithms. It is also shown that GAs as well can be seen

---

<sup>\*</sup>Mathematics and Modelling Institute of Montpellier, Montpellier University, 34095 Montpellier, France

<sup>†</sup>Stanford University, Microfluidic Laboratory, Stanford, USA

as discrete system of coupled ODEs. Therefore, the boundary value problem analysis has been applied to them as well improving their performances.

This global optimization problem is also complicated because the involved physic. It is multi-model in the sense that several partial differential equations need to be solved. Simulation of these equations being computational intensive, one also has to use low complexity approach in sensitivity and intermediate state calculations.

Previous application of control theory to the design of microfluidic devices based on simpler physics and only local minimization algorithms have been reported [6, 7, 8].

In section 2, we recall the state of the art on fast microfluidic protein folding devices design. Section 3 presents three global optimization algorithms with associated mathematical background. In Section 4 low-cost sensitivity evaluation approaches are presented. Section 5 introduces a short presentation of the physic of the problem and its numerical solution. Finally, section 6 shows and compares optimization results.

## 2 Fast Microfluidic Protein Folding devices

Microfluidic channel systems used in bio-analytical applications are fabricated using technologies derived from microelectronics industry including lithography, wet etching and bonding of substrates. Industrial applications of these techniques concern DNA sequencing, new drug molecules trials, pollution detection in water or food and protein folding [9].

Focusing on this last domain, important structural events occur on a microsecond time scale [10]. To resolve folding events of order 10 microseconds, mixer designs are required that effect mixing in a few microseconds or less. This can be performed, for instance, using photochemical initiation [11] and changes in temperature [12], pressure [13, 14] or chemical potential, as in salt or chemical denaturant concentration changes [15, 16, 17]. All these techniques provide perturbations of protein conformational equilibrium necessary to initiate folding. In comparison to temperature and pressure-jump relaxation technique, folding experiments based on changes in chemical potential, via rapid mixing of protein solutions into and out of chaotrope solvents, are more versatile. The technique is applicable to a wide range of proteins as most unfold reversibly in the presence of chemical denaturants such as urea and guanidine hydrochloride (GdCl) [15].

Until recently, the main limitation of mixer-based experiments was their inability to access very short timescales. Mixing time is ultimately limited by the time required for molecular diffusion across a finite length scale, and diffusion time scales as the square of diffusion length. Brody et al. [18] first proposed rapid mixers based on hydrodynamic focusing as a way to address the issue of reducing diffusion lengths under laminar flow conditions while minimizing sample consumption. Hydrodynamic focusing has been used to measure protein and RNA folding [19], with mixing times of a few hundreds of microseconds.

This paper discusses specific shape optimization for a new microfluidic mixer based on a continuous flow principle originally proposed by Knight et al. [20], and improved and demonstrated by Hertzog et al. [9]. The design leverages hydrodynamic focusing on the micron scale to reduce diffusion lengths [9]. This

mixer uses about eight orders of magnitude less labelled protein sample mass flow than a previously reported ultra-fast protein folding mixer [21], with flow rates of 3 nl/s and protein concentrations of tens of nanomolar.

### 3 Global optimization methods

We present three minimization methods: A typical genetic algorithm [22, 1], a new global optimization method based on the solution of boundary value problems, and an hybrid algorithm using ingredients from the two previous approaches.

#### 3.1 Genetic algorithm

Consider the minimization of a real functional  $J(x)$ , the so-called fitness function,  $x \in \Omega_{ad}$  is the optimization parameter and belongs to an admissible space  $\Omega_{ad}$  of dimension  $n_{dim}$ . Genetic algorithms approximate the global minimum (or maximum) of  $J$ , called the fitness function, through a stochastic process based on an analogy with the Darwinian evolution of species [1]:

A first family, called 'population',  $X^1 = \{(x)_l^1 \in \Omega_{ad}, l = 1, \dots, N_p\}$  of  $N_p$  possible solutions in  $\Omega_{ad}$  of the optimization problem, called 'individuals', is randomly generated in the search space  $\Omega_{ad}$ . Starting from this population  $X^1$  we build recursively  $N_{gen}$  new populations  $X^i = \{(x)_l^i \in \Omega_{ad}, l = 1, \dots, N_p\}$  with  $i = 1, \dots, N_{gen}$  through three stochastic steps:

We write  $X^i$  using the following  $(N_p, n_{dim})$ -matrix form:

$$X^i = \begin{bmatrix} x_1^i(1) & \dots & x_1^i(n_{dim}) \\ \vdots & \ddots & \vdots \\ x_{N_p}^i(1) & \dots & x_{N_p}^i(n_{dim}) \end{bmatrix} \quad (1)$$

At each iteration the following three steps are performed.

- **Selection:** each 'individual', representing a potential solution of the problem, is ranked with respect to the fitness function  $J$ . In this process, better individuals have higher chances to be chosen. They are called 'parents'.

Introducing  $\mathcal{S}$  a binary  $(N_p, N_p)$ -matrix with, for each line  $i$ , a value 1 on the  $j$ Th row when the  $j$ Th individual has been selected and 0 elsewhere

$$X^{n+1/3} = \mathcal{S}X^n \quad (2)$$

- **Crossover:** this process leads to a data exchange between two 'parents' and apparition of new individuals called 'child'.

- We introduce  $\mathcal{C}$  a real-valued  $(N_p, N_p)$ -matrix where for each couple of consecutive lines  $(2i - 1, 2i)$  ( $1 \leq i \leq \frac{N_p}{2}$ ), the coefficients of the  $l$ -Th and  $k$ -Th row are given by a  $2 \times 2$  matrices of the form

$$\begin{bmatrix} \lambda_1 & 1 - \lambda_1 \\ \lambda_2 & 1 - \lambda_2 \end{bmatrix} \quad (3)$$

In this expression,  $\lambda_1 = \lambda_2 = 1$  if no crossover is applied on the selected 'parents'  $l$  and  $k$  and are randomly chosen with a probability  $p_c$  in  $[0, 1]$  in the other case.

This step can be summarized as:

$$X^{n+2/3} = \mathcal{C}X^{n+1/3} \quad (4)$$

- **Mutation:** This process leads to new parameter values. For each 'child', we determine with a fixed probability  $p_m$  if their parameters should mute.

Introducing following families:

$$\mathcal{M}_+ = \{\mathcal{M}_{+i} \ i = 1, \dots, N_p\} \text{ and } \mathcal{M}_- = \{\mathcal{M}_{-j} \ j = 1, \dots, n_{dim}\} \text{ and } \{(\varepsilon_{i,j}) \ i = 1, \dots, N_p \ j = 1, \dots, n_{dim} / \varepsilon_{i,j} \in \mathbb{R}\}$$

with  $\mathcal{M}_{+i}$  a binary  $(N_p, N_p)$ -matrix which keep unchanged the  $i^{th}$  line of the matrix  $X^{n+2/3}$  and set to zero the other lines.  $\mathcal{M}_{+j}$  a  $(n_{dim}, n_{dim})$ -matrix which keep unchanged the  $j^{th}$  column of matrix  $X^{n+2/3}$  and set to zero the other columns.  $\varepsilon_{i,j}$  is equal to 1 if no mutation is applied and to (mutated value /  $(i, j)$  coefficient) otherwise.

In the same way, this step can be summarized as:

$$X^{n+1} = \sum_{i=1}^{N_p} \sum_{j=1}^{n_{dim}} \varepsilon_{i,j} \mathcal{M}_{+i} X^{n+2/3} \mathcal{M}_{-j} \quad (5)$$

Therefore, genetics algorithms can be seen as discrete dynamic systems:

$$X^{n+1} = \sum_{i=1}^{N_p} \sum_{j=1}^{n_{dim}} \varepsilon_{i,j} \mathcal{M}_{+i} (\mathcal{C}S X^n) \mathcal{M}_{-j} \quad (6)$$

this can be formally rewritten as:

$$X^{n+1} - X^n = \Lambda_2^n X^n \Lambda_3^n - X^n$$

which is a particular discretization of a set of nonlinear 1st order ODEs [23]:

$$\dot{X} = \Lambda_2(X) X \Lambda_3(X) - X \quad (7)$$

where the construction of  $\Lambda_i$  has been described above.

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 [1, 22].

Engineers like GAs because these algorithms do not require sensitivity computation, perform global and multi-objective optimization and are easy to parallelize. Their drawbacks remain their weak mathematical background, their computational complexity, their slow convergence and their lack of accuracy. The semi-deterministic algorithm (SDA) below aims to address some of these issues.

### 3.2 Semi-deterministic multi-level optimization

Most deterministic minimization algorithms, which perform the minimization of a function  $J : \Omega_{ad} \rightarrow \mathbb{R}$ , can be seen as discretizations of the following dynamical system [4, 24, 3] where  $x$  denotes the vector of control parameters belonging to an admissible space  $\Omega_{ad}$ .  $x_\zeta(\zeta) = \frac{dx(\zeta)}{d\zeta}$ .  $x_0 \in \Omega_{ad}$  is the initial condition.  $\zeta$  is a fictitious parameter.  $M$  is a local metric transformation and  $d$  a direction in  $\Omega_{ad}$ .

$$\begin{cases} M(\zeta)x_\zeta(\zeta) = -d(x(\zeta)) \\ x(\zeta = 0) = x_0 \end{cases} \quad (8)$$

For example if  $d = \nabla J$ , the gradient of the functional, and  $M = Id$ , the identity operator, we recover the classical steepest descent method while with  $d = \nabla J$  and  $M(\zeta) = \nabla^2 J(x(\zeta))$  the Hessian of  $J$ , we recover the Newton method. Quasi-Newton methods can also be recovered using approximate Hessian definition [25].

Theoretical background of the method requires the following assumptions [23]:

H1:  $J \in C^2(\Omega_{ad}, \mathbb{R})$ .

H2: the infimum  $J_m$  is known. This is often the case in industrial applications.

H3: the problem is admissible: the infimum is reached inside the admissible domain:  $\exists x_m \in \Omega_{ad}, s.t. J(x_m) = J_m$ .

H4:  $J$  is coercive (i.e.  $J(x) \rightarrow \infty$  when  $|x| \rightarrow \infty$ ).

Global optimization with system (8) is valid if the following system has a solution:

$$\begin{cases} M(\zeta)x_\zeta(\zeta) = -d(x(\zeta)) \\ x(0) = x_0 \\ J(x(Z_{x_0})) = J_m \quad \text{with finite } Z_{x_0} \in \mathbb{R} \end{cases} \quad (9)$$

This boundary value problem is over-determined (i.e. two conditions and only one derivative). The previous over-determination is an explanation of why we should not solve global optimization problems with methods which are particular discretizations of initial value problem for first order differential equations. We could use variants of classical methods after adding a second order derivative [4]:

$$\begin{cases} \eta x_{\zeta\zeta}(\zeta) + M(\zeta)x_\zeta(\zeta) = -d(x(\zeta)), \\ x(0) = x_0, \quad \dot{x}(0) = \dot{x}_0, \\ J(x(Z_{x_0})) = J_m \end{cases} \quad (10)$$

where  $x_{\zeta\zeta}(\zeta) = \frac{d^2x(\zeta)}{d\zeta^2}$ .

The over determination can be removed, for instance, by considering  $x_0 = v$  for (8) (resp.  $\dot{x}(0) = v$  for (10)) as a new variable to be found by the minimization of  $h(v) = J(x_v(Z_v)) - J_m$ , where  $x_v(Z_v)$  is the solution of (8) (resp. (10))

found at  $\zeta = Z_v$  starting from  $v$ .

The algorithm  $A_1(v_1, v_2)$  reads:

- $(v_1, v_2) \in \Omega_{ad} \times \Omega_{ad}$  given
- Find  $v \in \operatorname{argmin}_{w \in \mathcal{O}(v_2)} h(w)$  where  $\mathcal{O}(v_2) = \{t\overrightarrow{v_1 v_2}, t \in \mathbb{R}\} \cap \Omega_{ad}$
- return  $v$

The line search minimization might fail. For instance, a secant method degenerates on plateau and critical points. In this case, we add an external level to the algorithm  $A_1$ , keeping  $v_1$  unchanged, and looking for  $v_2$  by minimizing a new functional  $w \rightarrow h(A_1(v_1, w))$ .

This leads to the following two-level algorithm  $A_2(v_1, v_2)$ :

- $(v_1, v_2) \in \Omega_{ad} \times \Omega_{ad}$  given
- Find  $v' \in \operatorname{argmin}_{w \in \mathcal{O}(v_2)} h(A_1(v_1, w))$  where  $\mathcal{O}(v_2) = \{t\overrightarrow{v_1 v_2}, t \in \mathbb{R}\} \cap \Omega_{ad}$
- return  $v'$

The choice of initial conditions in this algorithm contains the non-deterministic feature of the algorithm. The construction can be pursued building recursively  $h^i(v_2^i) = \min_{v_2^i \in \Omega_{ad}} h^i(v_2^i) = h^{i-1}(A_{i-1}(v_1, v_2^i))$ , with  $h^1(v) = h(v)$  where  $i$  denotes the external level.

A one dimensional geometrical construction of the different functionals ( $J, h, h^2, h^3$ ) is shown in Figure 1 with  $J$  chosen to be non-convex,  $v_1$  fixed and  $v_2^i$  for  $i = 1, 2, 3$  take successively all the values of the discretized search space.  $h^2$  and  $h^3$  show growing attraction basins around the infimum. In that case, the attraction basin for  $h^3$  is the full search space.

Mathematical background for this approach and validation on academic test cases and solution of nonlinear PDEs are available [23, 3, 5, 26, 24, 27].

In practice, this algorithm succeeds if the trajectory passes close enough to the infimum (i.e. in  $B_\varepsilon(x_m)$  where  $\varepsilon$  defines the accuracy in the capture of the infimum). This means that we should consider for  $h$  a functional of the form

$$h(v) = \int_{T_1}^T (J(x_v(\tau)) - J_m)^2 d\tau, \quad \text{for } 0 < T_1 < \tau < T$$

where  $x_v(\tau)$  is the trajectory generated by (8) and  $T_1 = T/2$  for instance. Also, in the algorithm above,  $x_w(Z_w)$  is replaced by the best solution found over  $[0, Z_w]$ .

In cases where  $J_m$  is unknown, we set  $J_m = 0$  and look for the best solution for a given complexity and computational effort. This is the approach adopted here where we predefine the effort we would like to make in each level of the algorithm.

### 3.3 Hybridization

It is interesting to notice that once GA is seen as a dynamical system (7) for the population, it can be used as core optimization method in SDA. We call this approach HGSA (hybrid genetic/semi-deterministic algorithm). The aim here is to find a compromise between the robustness of GAs and low-complexity features of SDA.

In practice, as final convergence is difficult with GA based algorithms, one should always complete GA iterations by a descent method for better accuracy. This is useful especially when the functional is flat around the infimum.

### 3.4 Academic test case

SDA, HGSA and GA algorithms have been compared on the Rastrigin function given by:

$$J(x) = \sum_{n=1}^2 (x_n^2 - \cos(18x_n))$$

with  $x \in [-5, 5]^N$  and  $N = 10, 100, 1000$ . The infimum is 0 reached at the origin.

The two-level SDA algorithm  $A_2(v_1, v_2)$  is used with  $(v_1, v_2)$  chosen randomly in  $\mathbb{R}^N \times \mathbb{R}^N$  and the dynamical system corresponding to the steepest descent method as core optimization algorithm [3].

HGSA and GA are applied with the following parameters for the three associated stochastic processes (see section 3.1):

- The population size has been set to  $N_p = 180$  (resp. 10) for GA (resp. HGSA).
- The selection is a roulette wheel type proportional to the rank of the individual 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 probability of  $p_m = 0.15$ .
- A one-elitism principle, that consists in keeping the current best individual in the next generation, has also been imposed.

All these parameters are fixed and used in all computations of this paper. These values give a good compromise between computational complexity and accuracy [5, 23]. For this test case, the generation number  $N_{gen}$  is unbounded for GA and set to 10 for HSGA.

Results are presented on Table 1. The SDA algorithm is faster and more efficient than GA and HGSA on this case. However the HGSA give a good compromise to the GA. This has also been observed, with other sets of parameters, on other analytical cases available in the literature [5, 23].

## 4 Low-cost sensitivity

Consider a general simulation loop, leading from shape parametrization  $x$  to the cost functional  $J$ :

$$J(x) : x \rightarrow q(x) \rightarrow U(q(x)) \rightarrow J(x, q(x), U(q(x))) \quad (11)$$

where  $q$  is the shape geometry and  $U$  is the state equation solution.

The Jacobian of  $J$  is given by:

$$\frac{dJ}{dx} = \frac{\partial J}{\partial x} + \frac{\partial J}{\partial q} \frac{\partial q}{\partial x} + \frac{\partial J}{\partial U} \frac{\partial U}{\partial q} \frac{\partial q}{\partial x} \quad (12)$$

The last term  $\frac{\partial J}{\partial U} \frac{\partial U}{\partial q} \frac{\partial q}{\partial x}$  is the more expensive to compute as it requires the linearization of the state equations.

One way to reduce computational effort of sensitivity evaluations is to use reduced complexity models to provide an inexpensive approximation of the last term in [12]. Consider a reduced model  $\tilde{U}(x) \sim U(q(x))$ . The incomplete gradient of  $J$  with respect to  $x$  can be improved evaluating the former term in (12) linearizing this model. Note that  $\tilde{U}$  is never used in the definition of the state  $U$ , but only in the approximation  $\frac{\partial \tilde{U}}{\partial q} \frac{\partial q}{\partial x}$ . More precisely, we linearize the following approximate simulation loop

$$x \rightarrow q(x) \rightarrow \tilde{U}(x) \frac{U(q(x))}{\tilde{U}(x)} \quad (13)$$

freezing  $U(q(x))/\tilde{U}(x)$  which gives

$$\frac{dJ}{dx} \approx \frac{\partial J(U)}{\partial x} + \frac{\partial J(U)}{\partial q} \frac{\partial q}{\partial x} + \frac{\partial J(U)}{\partial U} \frac{\partial \tilde{U}}{\partial x} \frac{U(q(x))}{\tilde{U}(x)}. \quad (14)$$

A simple example shows the importance of the scaling introduced in [13]. Consider  $U(x) = \log(1+x)$  scalar for simplicity and  $J(x) = U^2(x)$  with  $dJ(x)/dx = 2U(x)U'(x) = 2\log(1+x)/(1+x) \sim 2\log(1+x)(1-x+x^2\dots)$  and consider  $\tilde{U}(x) = x$  as the reduced complexity model, valid around  $x=0$ . To see the impact of the scaling factor we compare  $J'(x) \sim 2U(x)\tilde{U}'(x) = 2\log(1+x)$  with  $J'(x) \sim 2U(x)\tilde{U}'(x)(U(x)/\tilde{U}(x)) = 2\log(1+x)(\log(1+x)/x) \sim 2\log(1+x)(1-x/2+x^2/3\dots)$ .

Another way to define low-complexity models is to use a different level of discretization for  $U$  with the same state equation. We can look for state sensitivity on coarse meshes while the state is evaluated on much finer discretizations:

$$\frac{dJ}{dx} = \frac{\partial J}{\partial x}(U_f, q_f) + \frac{\partial J}{\partial q} \frac{\partial q}{\partial x}(U_f, q_f) + \frac{\partial J}{\partial U} \frac{\partial U}{\partial q} \frac{\partial q}{\partial x}(I_f^c(U_f), q_c)$$

where  $f$  and  $c$  subscripts denote fine and coarse meshes,  $U_f$  denote the state equation solution evaluated on  $f$ ,  $I_f^c(\cdot)$  is an interpolation operator between the fine and coarse meshes. By fine mesh we mean a mesh enough fine for the solution to be mesh independent. This means that the linearization is performed on a coarse mesh, however around an accurate state variable computed on a fine mesh. In that case, obviously if the coarse mesh tends to the fine one, the



approximate gradient tends to the gradient on the fine mesh. In addition, to the two levels of refinements used for state and sensitivity calculations, state evaluations for gradient calculation can be only made partially. Hence, only partial convergence of solver method in the solution of the state equations is required starting from  $I_f^c(U_f)$ . This corresponds to the fact that the semi-deterministic algorithm above only needs a descent direction  $d$  such that  $d \cdot \nabla J > \varepsilon > 0$  (see Figure 2) [23, 5, 3].

This last approach is easy to couple with any commercial code.

## 5 Fast-micro-mixer modelling

The SDA, HGSA and GA algorithms above are used to optimize the shape of a given fast-micro-mixer in order to reduce its mixing time. There are no general definitions of mixing time and several definitions of the degree of mixing exist in the literature [28]. In most cases, the best definition of mixing time is an ad hoc rule that takes into account the figures of merit of a specific application. In section 5.3, we propose a mixing definition for our mixer that well characterizes the temporal resolution of subsequent macromolecular folding kinetics measurements.

### 5.1 Shape design

The mixer shape considered is a typical three-inlet/single-outlet channel architecture proposed by Knight [20] (see Figure 3). Due to the fact that our model is symmetric we only study half of the mixer [9] (see Figure 4-*Left*). Our model is a 2D approximation of the physical system [29]. Experiments show a 5 percent deviation from a 3D modelling which is satisfactory for a 2D model to be used as low-complexity model in optimization [9].

Our aim is to optimize the corner shapes. We parameterize the corner regions by cubic splines (see Figure 4-*Right*). The total number of parameters is 8, 4 for each corner. In addition, we account for the following constraints:

- The considered fast-Micro-Mixer is  $22\mu\text{m}$  long and  $10\mu\text{m}$  large.
- The lithography step in fabrication limits the minimum feature size to a minimum of 1 to 2  $\mu\text{m}$ .
- The width of the side channel nozzles is set to 3  $\mu\text{m}$  and the width of the center channel nozzles to 2  $\mu\text{m}$  to mitigate clogging issues.
- The depth of the channels is set to  $\sim 10 \mu\text{m}$  to optimize the fluorescence signal with a confocal system. In addition, it is difficult to etch deeper on fused silica [9].
- The physical properties of buffers and guanidine hydrochloride denaturant used here for protein folding studies have known parameters such as density, viscosity, and diffusivity.
- Finally, the maximum side velocity is  $U_s = 10^{-4} \text{m/s}$ . Hence, a typical flow Reynolds number based on sides channel thickness and flow inlet is  $Re_w = \frac{U_s w_s}{\eta} \sim 15$ .

- The device length scales required for the continuum assumption to hold are different for the flow of liquids and gases [30]. For gases, the appropriate length scale is typically the mean free path of the gas. In liquids, molecules are tightly packed and we use the characteristic size of the molecule. For water at standard conditions, the molecular diameter is  $310^{-10}m$  which justify the continuum assumption for this problem as it gives a Knudsen number of order  $10^{-4}$ .

Thus, the corresponding search space of the optimization problem is  $\Omega = [x_i^{min}, x_i^{max}]_{i=1}^8$  where  $x_i^{min}$  (resp.  $x_i^{max}$ ), the minimum (resp. maximum) value of the  $i^{th}$  parameter, are fixed by the previous constraints.

## 5.2 State equations

The mixer flow was analyzed using numerical solutions of the full Navier-Stokes fluid flow equations and a convection diffusion equation describing concentration fields  $c$  of the guanidine hydrochloride denaturant. Only steady configurations have been considered as we are not interested in the behavior of the device during its transient set up.

These flow simulations were used to explore the guanidine hydrochloride performance of a variety of mixer designs with systematically varied flow and geometric parameters. The model is applied to mixer shape designs described in 5.1. Mixer flow and concentration field  $c$  are computed using the following system of equations:

$$\begin{cases} -\nabla \cdot (\eta(\nabla u + (\nabla u)^T)) + \rho(u \cdot \nabla)u + \nabla p = 0 \\ \nabla \cdot u = 0 \\ \nabla \cdot (-D\nabla c + cu) = 0 \end{cases} \quad (15)$$

where  $(u, p)$  is the flow velocity vector and pressure field,  $\rho = 1013kg/m$  is the density,  $\eta = 8.10^{-4}Pa.s$  the dynamic viscosity and  $D = 2^{-9}m^2/s$  is the diffusion coefficient.

Finally, the following boundary conditions are assumed:  $u = 3.210^{-4}m/s$  on side inlets,  $u = 3.210^{-6}m/s$  on center inlet,  $u \cdot t = 0$  on the exit,  $u \cdot n = 0$  on the center symmetry line,  $u = 0$  elsewhere on the shape border.  $(t, n)$  is the local orthonormal reference frame along the boundary.  $c$  is prescribed at inlet and normal zero gradient is assumed for all other boundaries.  $c = 0$  at side inlet and  $c = 1$  at center inlet.

The Incompressible Navier-Stokes equations are solved iteratively. Velocity and pressure spacial discretization is based on P2-P1 Lagrange finite elements to stabilize to realize the Ladyzhenskaya, Babuska and Brezzi (LBB) stability condition. The convective diffusion equation is solved using a streamline-upwind/Petrov-Galerkin (SUPG) method [31]. A direct damped Newton method is then used to solve the nonlinear system coming from 15 [32].

## 5.3 Cost Function

The cost function to minimize is the mixing time of the considered Lagrangian fluid particle travelling along the centerline. The mixing time is defined as the time required to change local concentration from 90% to 30% of the inlet value

$c_0$ :

$$J(x) = \int_{y_{90}}^{y_{30}} \frac{dy}{v} \quad (16)$$

Where  $y_{90}$  and  $y_{30}$  denote respectively the points along the symmetry line where the concentration is at 90% and 30% of  $c_0$ .  $v$  is the normal component of the velocity.

This modelling has been validated by a posteriori prototyping [9]. We are interested by an ensemble of state equations enough rich for the optimization problem to be valid but also as simple as possible to control the computational complexity. In particular, we need to keep the cost of state evaluations low for genetic algorithms and also for sensitivity analysis.

## 6 Results and discussion

In GA (resp. HGSA) the maximum generation number is set to 30 (resp 10). They start from a random initial population  $X^1$ . SDA starts from an initial shape made with a smoothed 90 degrees corners parameterized with splines to keep the admissible regularity. The SDA descent direction is  $d = \nabla J$ . In three optimizations, the mixing time has been decreased from  $8\mu s$  to  $1.15\mu s$  (see Figure 6-*Right*). Initial and optimized shapes are presented in Figure 6-*Left*.

For GA (resp. HGSA) the total number of functional evaluations is 5400 (resp. 2500). For SDA the evaluation number is 3500 with more than 90 % of the evaluations on a coarse mesh with incomplete state evaluations (Figure 5). The cost of an incomplete evaluation of the gradient is around two evaluations of the functional on a fine mesh. Convergence histories are given in Figure 7. As we can see on this Figure, SDA has visited several attraction basins before exploring the best element basin. Each evaluation on the fine (resp. coarse) level requires about one minute (resp. 20s) on a 3Ghz PC computer. Hence, GA requires about 4 days, HGSA 2 days and SDA less than 4 hours to reach the same minimum. In addition, with SDA the infimum is reached sooner in the optimization (see Figure 7).

## 7 conclusion

An unified formulation for deterministic and stochastic global optimization based on the solution of initial and boundary value problems for dynamics systems has been presented. The solution of this boundary value problem leads to a recursive semi-deterministic minimization algorithm where non-deterministic aspects is reduced as much as possible to limit the complexity of the method. To keep the computational complexity low and make the problem easy to solve with industrial softwares approximate gradient evaluation has been used on coarse meshes.

Both algorithms over-perform in term of computational complexity genetic algorithms when applied to the academic configurations and for the design of a fast-micro-mixer.

Preliminary measurements performed using the mixer designs described here show that the optimized mixers have significant improvements on mixing time. Mixing time was shown to be decreased from 7 to 3  $\mu s$ , and the comparisons

between experiments and data are within the uncertainty of the measurements [33].

## References

- [1] D. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, 1989.
- [2] C. M. Fonseca and J. Fleming. An overview of evolutionary algorithms in multi-objective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.
- [3] B. Mohammadi and J-H. Saiac. *Pratique de la simulation numérique*. Dunod, 2002.
- [4] H. Attouch and R. Cominetti. A dynamical approach to convex minimization coupling approximation with the steepest descent method. *J. Differential Equations*, 128(2):519–540, 1996.
- [5] B. Ivorra, B. Mohammadi, and P. Redont. Low-complexity global optimization by solution of bvp. *Journal of Global Optimization*, submitted, 2005.
- [6] B. Mohammadi, J. Santiago, and J. Molho. Incomplete sensitivities in the design of minimal dispersion fluidic channels. *Comput. Methods Appl. Mech. Engrg.*, 192:4131–4145, 2003.
- [7] B. Mohammadi, R. Bharadwaj, and J. Santiago. Design and optimization of on-chip capillary electrophoresis. *Electrophoresis Journal*, 23(16):2729–2744, 2002.
- [8] B. Mohammadi, J. Molho, A. Herr, J. Santiago, T. Kenny, R. Brennen, and Gordon G. Optimization of turn geometries for on-chip electrophoresis. *Analytical Chemistry*, 73(6):1350–1360, 2001.
- [9] D.E. Hertzog, X. Michalet, M. Jager, X. Kong, J.G. Santiago, S. Weiss, and O. Bakajin. Femtomole mixer for microsecond kinetic studies of protein folding. *Analytical Chemistry*, 75(24):7169–7178, 2004.
- [10] Roder H. Stepwise helix formation and chain compaction during protein folding. *Proceedings of the National Academy of Sciences of the USA*, 101:1793–1794, 2004.
- [11] C. M. Jones, E. R. Henry, Y. Hu, C.-K. Chan, S. D. Luck, A. Bhuyan, H. Roder, J. Hofrichter, and W. A. Eaton. Fast events in protein folding initiated by nanosecond laser photolysis. *Proceedings of the National Academy of Sciences of the USA*, 90:11860–11864, 1993.
- [12] S. J. Hagen and W. A. Eaton. Two-state expansion and collapse of a polypeptide. *Journal of Molecular Biology*, 301:1037–1045, 2000.
- [13] K. M. Pryse, T. G. Bruckman, B. W. Maxfield, and E. L. Elson. Kinetics and mechanism of the folding of cytochrome c. *Biochemistry*, 31(22):5127–5136, 1992.

- [14] M. Jacob, G. Holtermann, D. Perl, J. Reinstein, T. Schindler, M. A. Geeves, and F. X. Schmid. Microsecond folding of the cold shock protein measured by a pressure jump technique. *Biochemistry*, 38:2882–2891, 1999.
- [15] C.K. Chan, Y. Hu, S. Takahashi, D. L. Rousseau, and W. A. Eaton. Submillisecond protein folding kinetics studied by ultrarapid mixing. *Proceedings of the National Academy of Sciences of the USA*, 94:1779–1784, 1997.
- [16] Pollack L., M. W. Tate, N. C. Darnton, J. B. Knight, S. M. Gruner, W. A. Eaton, and R. H. Austin. Compactness of the denatured state of a fast-folding protein measured by submillisecond small angle x-ray scattering. *Proceedings of the National Academy of Sciences of the USA*, 96:10115–10117, 1999.
- [17] S.-H. Park, M. C. R. Shastry, and H. Roder. Folding dynamics of the b1 domain of protein g explored by ultrarapid mixing. *Nature, Structural Biology*, 6:943–947, 1999.
- [18] J. P. Brody, P. Yager, R.E. Goldstein, and R.H. Austin. Biotechnology at low reynolds numbers. *Biophysical Journal*, 71(6):3430–3441, 1996.
- [19] R. Russell, I.S. Millet, M.W. Tate, L.W. Kwok, B. Nakatani, S.M. Gruner, S.G.J. Mochrie, V.S. Pande, S. Doniach, D. Herschlag, and L. Pollack. Rapid compaction during rna folding. *Proceedings of the National Academy of Sciences of the USA*, 99:4266–4271, 2002.
- [20] J. B. Knight, A. Vishwanath, J. P. Brody, and R. H. Austin. Hydrodynamic focusing on a silicon chip: Mixing nanoliters in microseconds. *Physical Review Letters*, 80:3863–3866, 1998.
- [21] M. C. R. Shastry, S. D. Luck, and H. Roder. A continuous-flow capillary mixer to monitor reactions on the microsecond time scale. *Biophysical Journal*, 74:2714–2721, 1998.
- [22] L. Dumas, V. Herbert, and F. Muyl. Hybrid method for aerodynamic shape optimization in automotive industry. *Computers and Fluids*, 33(5):849–858, 2004.
- [23] B. Ivorra. Semi-deterministic global optimization. *PhD. University of Montpellier 2*, 2006.
- [24] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2001.
- [25] G.N. Vanderplaats. *Numerical optimization techniques for engineering design*. Mc Graw-Hill, 1990.
- [26] L. Debiane, B. Ivorra, B. Mohammadi, F. Nicoud, A. Ern, T. Poinsot, and H. Pitsch. Temperature and pollution control in flames. In *Proceeding of the Summer Program*, pages 367–375, Center for Turbulence Research, NASA AMES/Stanford University, USA, 2004.
- [27] B. Mohammadi and O. Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics.*, 36:255–279, 2004.

- [28] J. M. Ottino. *The Kinematics of Mixing: Stretching, Chaos, and Transport*. Cambridge University Press, 1989.
- [29] N. Darnton, O. Bakajin, R. Huang, B. North, J. Tegenfeldt, E. Cox, J. Sturn, and R. H. Austin. Condensed matter. *Journal of Physics*, 13:4891–4902, 2001.
- [30] W. M. Deen. *Analysis of Transport Phenomena*. New York, Oxford University Press, 1998.
- [31] T. Hughes and A. Brooks. A multi-dimensional upwind scheme with no crosswind diffusion, in t. hughes, ed., finite element methods for convection dominated flows. *ASME, New York*, 34:19–35, 1979.
- [32] P. Deuffhard. A modified newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numer. Math.*, 32:289–315, 1974.
- [33] D.E. Hertzog, B. Ivorra, B. Mohammadi, O. Bakajin, and J.G. Santiago. Optimization of fast microfluidic mixers for protein folding. *in preparation*.

	SDA	GA	HGSA
N=10	66	6000	2600
N=100	70	$O(1e5)$	$O(1e4)$
N=1000	80	$O(1e7)$	$O(1e5)$

Table 1: Rastrigin function. Total number of functional evaluations needed to reach the infimum with an accuracy of  $10^{-6}$ .

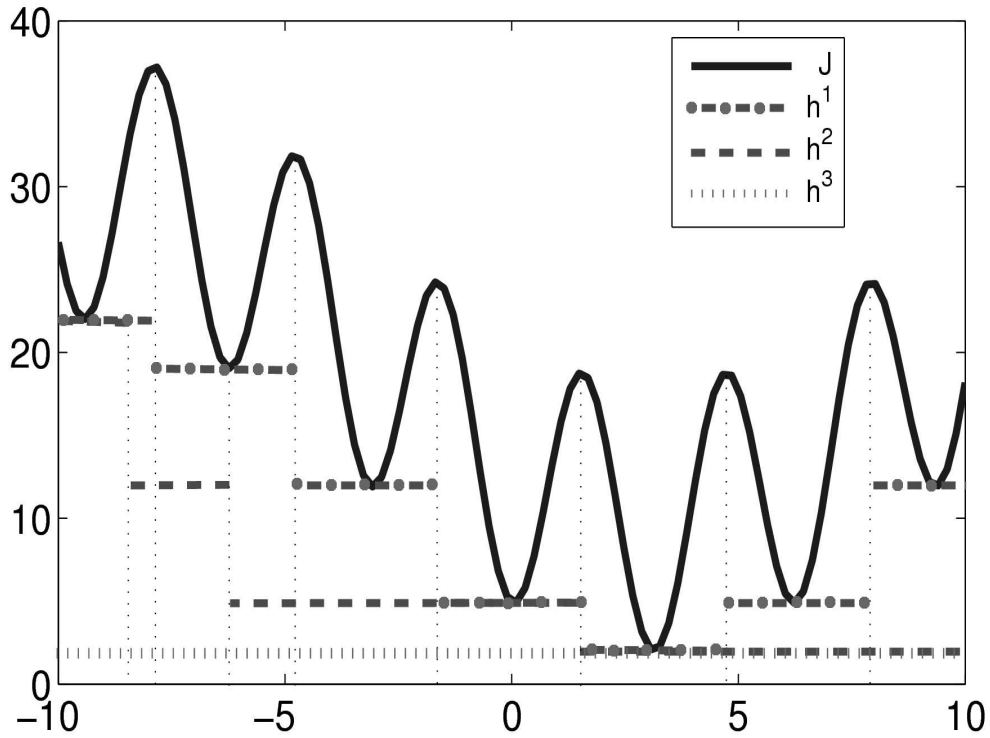


Figure 1:  $J(x)$  (continuous line) chosen non convex,  $h^1$  (.),  $h^2$  (-) and  $h^3$  (..) built on a uniform sampling of the control space  $x \in [-10, 10]$  at each level of the algorithm.

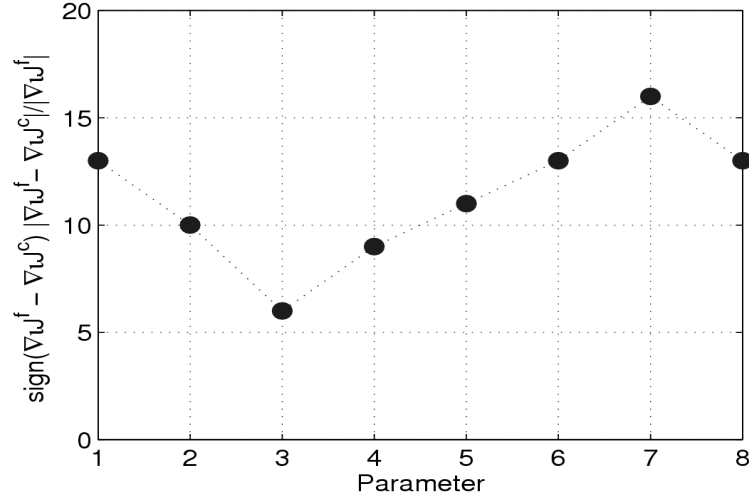


Figure 2: Comparison of gradients computed on the fine  $f$  vs. coarse  $c$  meshes. Average value of  $(\text{sign}(\nabla_i J^f - \nabla_i J^c) \frac{|\nabla_i J^f - \nabla_i J^c|}{|\nabla_i J^f|})$  where  $i=1, \dots, 8$  denotes the parameter number. We notice that the sign is always correct.

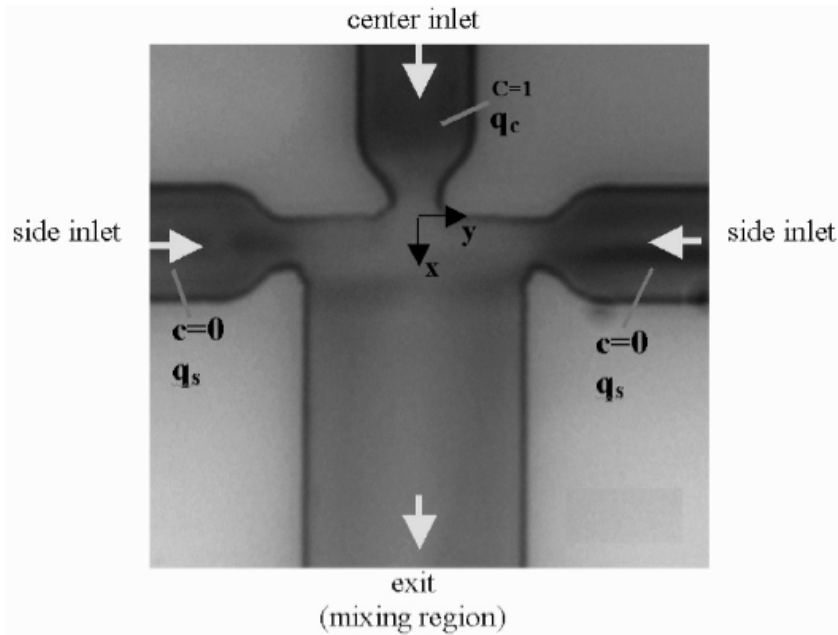


Figure 3: Typical fast-micro-mixer geometry.  $q_s$  and  $q_c$  are respectively the side/center injection velocities.  $c$  is the denaturant concentration.



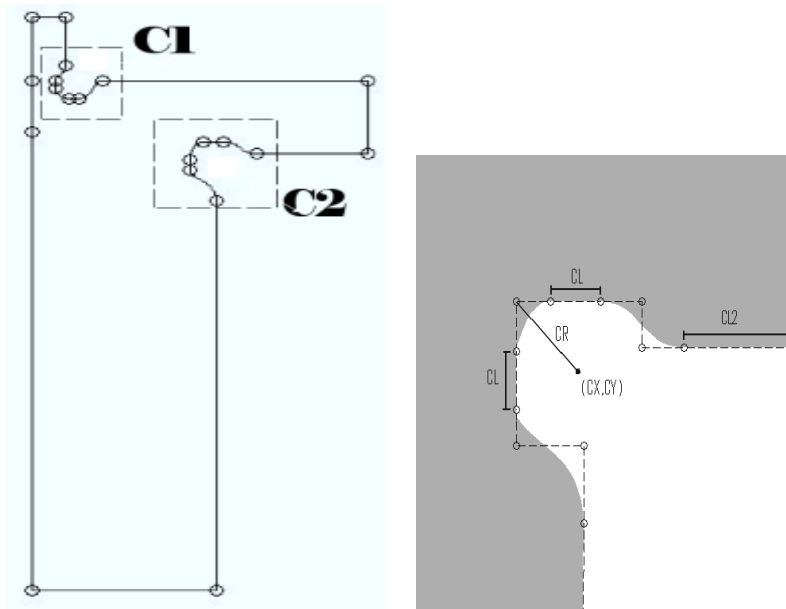


Figure 4: **Left:** Half-Shape parameterization. The corners are denoted by  $C_1$  and  $C_2$ . **Right:** Typical parameterization of a corner. Here  $C_2$ . We consider 4 parameters:  $C_x, C_r, C_l, C_{l2}$ .  $C_y$  is fixed.

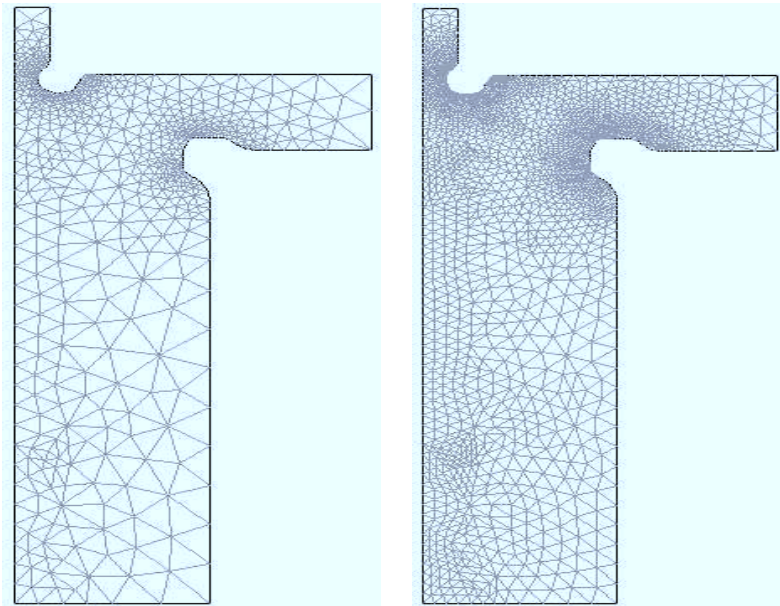


Figure 5: **Left:** For gradient calculation the state is partially evaluated on a coarse mesh. **Right:** On each new shape the state is calculated on a fine mesh.

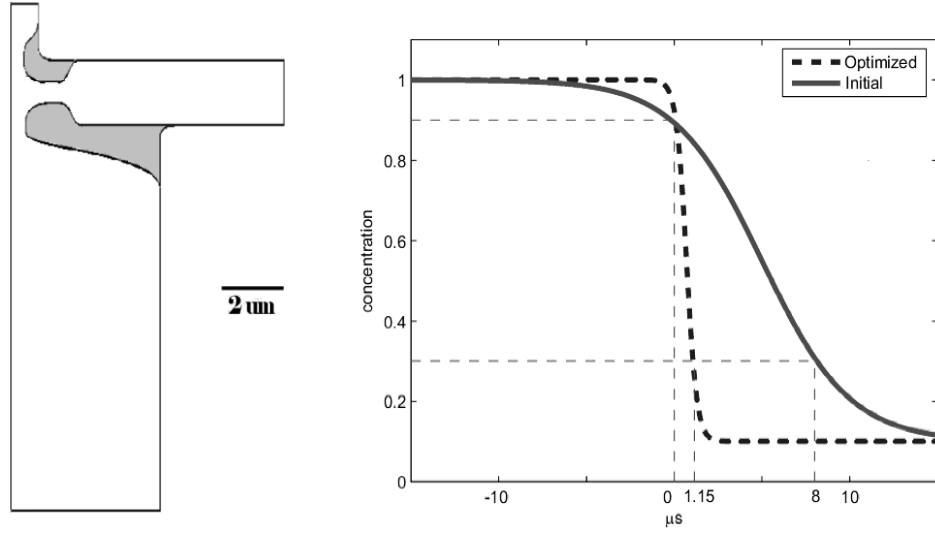


Figure 6: **Left:** SDA and GA optimized shape superposed over Initial shape. Parts in grey have been removed by the algorithms. **Right:** Concentration evolution for the initial and Optimized shapes.

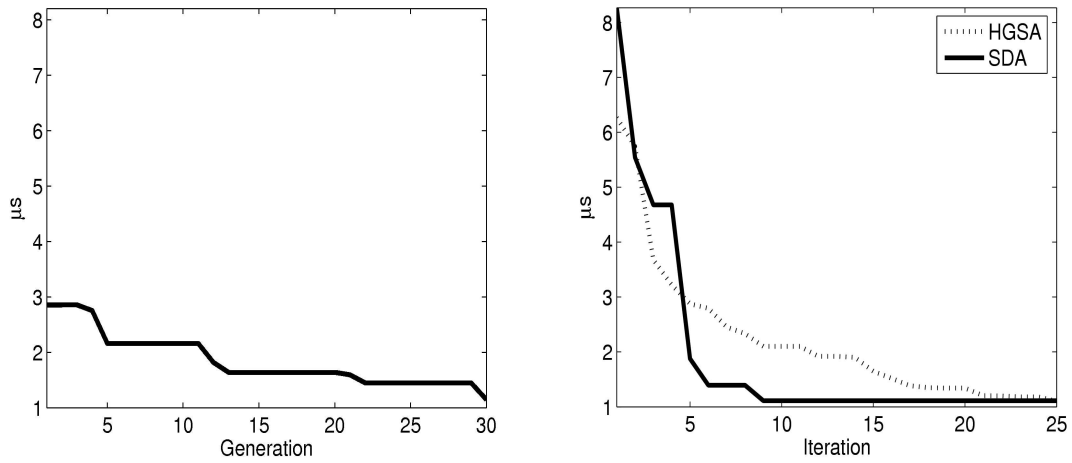


Figure 7: **Left:** Best element convergence history vs. generation iterations for GA. **Right:** Best element convergence history vs. iterations for SDA and HGSA.