# Comparison of Global Optimization Methods for Drag Reduction in the Automotive Industry

Laurent Dumas[1], Vincent Herbert, and Frédérique Muyl[2]

[1] Laboratoire Jacques-Louis Lions,
Université Pierre et Marie Curie,
75252 Paris Cedex 05, France
dumas@ann.jussieu.fr,
[2] PSA Peugeot Citroën
2 route de Gisy, 78943 Vélizy Villacoublay, France
{vincent.herbert, frederique.muyl}@mpsa.com

**Abstract.** Various global optimization methods are compared in order to find the best strategy to solve realistic drag reduction problems in the automotive industry. All the methods consist in improving classical genetic algorithms, either by coupling them with a deterministic descent method or by incorporating a fast but approximated evaluation process. The efficiency of these methods (called HM and AGA respectively) is shown and compared, first on analytical test functions, then on a drag reduction problem where the computational time of a GA is reduced by a factor up to 7.

## 1 Introduction

The topic of drag reduction in the automotive industry has been extensively studied since many years because of its great importance in terms of fuel consumption reduction. However, a computational and automatic approach of this problem has been unreachable until recently because of its difficulty due to the main two reasons: the complexity of the cost function to minimize and the computation time of each evaluation. The first attempt in this direction has been presented, up to our knowledge, by the present authors in [1]: this article was describing the drag minimization of a simplified 3d car shape with a global optimization method that coupled a genetic algorithm and a second order BFGS method. The interest of such hybrid method had been clearly shown on analytic cases where the convergence speed up was spectacular compared to a classical genetic algorithm. Unfortunately, the improvement for the industrial case was not so important because of the lack of accuracy of the gradient computation and so forth of the hessian approximation. The present article is intended to go further in this direction by performing a large comparison of different strategies to enhance the convergence of genetic algorithms, either with a hybrid method or with an approximated evaluation process. All these global optimization methods, described in paragraph 2, are compared on two analytic cases in paragraph 3 and on an industrial problem of car drag reduction in paragraph 4.

O. Gervasi et al. (Eds.): ICCSA 2005, LNCS 3483, pp. 948–957, 2005.

## 2 Global Optimization Methods

There exists many methods for minimizing a cost function $J$ defined from a set $\mathcal{O} \subset \mathbb{R}^n$ to $\mathbb{R}_+$. Among them, the class of genetic algorithms (GA), which main principles are recalled in the next subsection, has the major advantage to seek for a global minimum. Unfortunately, this method is very time consuming because of the large number of cost function evaluations that are needed. All the hybrid optimization methods presented in subsection 2.2 greatly reduce this time cost by coupling a GA with a deterministic descent method. Another way to speed up the convergence of a GA is described in subsection 2.3 and consists in doing fast but approximated evaluations during the optimization process. In this last case, an improvement of an existing approximation model is proposed.

### 2.1 Genetic Algorithms (GA)

Genetic algorithms are global optimization methods directly inspired from the Darwinian theory of evolution of species ([2]). They consist in following the evolution of a certain number $N_p$ of possible solutions, $(x_i)_{1 \leq i \leq N_p} \in \mathcal{O}^{N_p}$, also called population. To each element (or individual) $x_i$ of the population is affected a fitness value inversely proportional to $J(x_i)$ in case of a minimization problem. The population is regenerated $N_g$ times by using three stochastic principles called selection, crossover and mutation, that mimic the biological law of the 'survival of the fittest'. These principles are applied in the following way: at each generation, $\frac{N_p}{2}$ couples are selected by using a roulette wheel process with respective parts based on the fitness rank of each individual in the population. To each selected couple, the crossover and mutation principles are then successively applied with a respective probability $p_c$ and $p_m$. The crossover of two elements consists in creating two new elements by doing a barycentric combination of them with random and independent coefficients in each coordinate. The mutation principle consists in replacing a member of the population by a new randomly chosen in its neighborhood. A one-elitism principle is added in order to be sure to keep in the population the best element of the previous generation. Thus, the algorithm writes as:

```
– Randomly choose the initial population $P_1 = \{x_{i,1} \in \mathcal{O}, 1 \leq i \leq N_p\}$
– $n_g = 1$. Repeat until $n_g = N_g$
–     Evaluate $\{J(x_{i,n_g}), 1 \leq i \leq N_p\}$ and $m = \min\{J(x_{i,n_g}), 1 \leq i \leq N_p\}$
–     1-elitism: if $n_g \geq 2$ & $J(X_{n_g-1}) < m$ then $x_{i,n_g} = X_{n_g-1}$ for a random $i$
–     Affect a fitness value to each element. Call $X_{n_g}$ the best element
–     for $k$ from 1 to $\frac{N_g}{2}$
–         Selection of $(x_{\alpha,n_g}, x_{\beta,n_g})$ with respect to the fitness value
–         with probability $p_c$: replace $(x_{\alpha,n_g}, x_{\beta,n_g})$ by $(x'_{\alpha,n_g}, x'_{\beta,n_g})$ by crossover
–         with probability $p_m$: replace $(x'_{\alpha,n_g}, x'_{\beta,n_g})$ by $(x''_{\alpha,n_g}, x''_{\beta,n_g})$ by mutation
–     end for
– $n_g = n_g + 1$. Generate the new population $P_{n_g}$
```

### 2.2 Hybrid Methods (HM)

The principle of hybrid optimization has been introduced in the previous decade ([3],[4],[5]) in order to improve the convergence speed of an evolutionary algorithm, such as a genetic algorithm, in the case of computationally expensive optimization problems under limited computational budget. The general idea is to couple a GA with a deterministic descent method which will explore more rapidly the local minima of the objective function. More recently, this idea has been used in conjunction with computationally cheap surrogate approximation models (see the next paragraph for more details). In the present approach, the local search is only performed starting from the best current element $X_{n_g}$ at a generation $n_g$ after a stagnation in the GA process has been observed during $N_{stag} \in \mathbb{N}$ generations. Moreover, this procedure is done with exact evaluations of the cost function. Thus, the algorithm of the previous subsection is modified in the following way:

- (*Affect a fitness value to each element. Call* $X_{n_g}$ *the best element* )
- *if* $X_{n_g-N_{stag}-1} \neq X_{n_g}$ *and* $X_{n_g-N_{stag}} = ... = X_{n_g}$ *then apply a descent method starting from* $X_{n_g}$
- ( *for k from* 1 *to* $\frac{N_g}{2}$ )

Such hybrid method is called $\text{HM}_{N_{stag}}$. Note in particular that $\text{HM}_0$ consists to apply a descent method to the current best element of the population after each improvement in the GA process. In [1], a hybrid method of type $\text{HM}_3$ has been used.

The descent method can be of any type, first or second order (as in [1] when the BFGS algorithm has been used). In the present article, a first order gradient method with a backtracking line search strategy has been selected in order to limit the influence of the approximation eventually done in the gradient evaluation. A maximal number of five iterations in the descent method has also been fixed.

### 2.3 Genetic Algorithms with Approximated Evaluations (AGA)

Another idea to speed up the GA convergence when the computation time of the cost function $x \mapsto J(x)$ is high, is to take benefit of the large and growing data base of exact evaluations by making fast and approximated evaluations $x \mapsto \tilde{J}(x)$ leading to what is called surrogate or meta-models (see [6] and [7] for an overview). This general idea has also been used recently in conjunction with a hybrid process: in [8], a strategy for coupling an evolutionary algorithm with local search and quadratic response surface methods is proposed whereas a parallel hybrid evolutionary algorithm framework that leverages surrogate models for solving computationally expensive design problems with general constraints is presented in [9] and further extended. In the present work, the surrogate model is developed independently of the hybrid process and consists to perform exact evaluations only for all the best fitted elements of the population (in the sense of $\tilde{J}$) and for one randomly chosen element. The new algorithm, called AGA is thus deduced from the algorithm of section 2.1 and writes as:

```
– ( $n_g = 1$. Repeat until $n_g = N_g$ )
– if $n_g = 1$ then evaluate $\{J(x_{i,n_g}), 1 \leq i \leq N_p\}$
– elseif $n_g \geq 2$
–     for i from $1$ to $N_p$
–         Evaluate $\tilde{J}(x_{i,n_g})$.
–         if $\tilde{J}(x_{i,n_g}) < J(X_{n_g-1})$ then evaluate $J(x_{i,n_g})$
–     end for
–     for a random i: evaluate $J(x_{i,n_g})$
– end elseif
– ( 1-elitism )
```

The interpolation method chosen here comes from the field of neural networks and is called RBF (Radial Basis Function) interpolation ([10]). Suppose that the function $J$ is known on $N$ points $\{T_i, 1 \leq i \leq N\}$, the idea is to approximate $J$ at a new point $x$ by making a linear combination of radial functions of the type:

$$\tilde{J}(x) = \sum_{i=1}^{n_c} \psi_i \Phi(||x - \hat{T}_i||)$$

where:

- $\{\hat{T}_i, 1 \leq i \leq n_c\} \subset \{T_i, 1 \leq i \leq N\}$ is the set of the $n_c \leq N$ nearest points to $x$ for the euclidian norm $||.||$, on which an exact evaluation of $J$ is known.
- $\Phi$ is a radial basis function chosen in the following set:

$$\begin{aligned} \Phi_1(u) &= \exp(-\frac{u^2}{r^2}) \\ \Phi_2(u) &= \sqrt{u^2 + r^2} \\ \Phi_3(u) &= \frac{1}{\sqrt{u^2 + r^2}} \\ \Phi_4(u) &= \exp(-\frac{u}{r}) \end{aligned}$$

for which the parameter $r > 0$ is called the attenuation parameter.

The scalar coefficients $(\psi_i)_{1 \leq i \leq n_c}$ are obtained by solving the least square problem of size $N \times n_c$:

$$\text{minimize} \quad err(x) = \sum_{i=1}^{N} (J(T_i) - \tilde{J}(T_i))^2 + \lambda \sum_{j=1}^{n_c} \psi_j^2$$

where $\lambda > 0$ is called the regularization parameter.

In order to attenuate or even remove the dependancy of this model to its attached parameters, a secondary global optimization procedure (namely a classical GA) has been over-added in order to determine for each $x$, the best values (with respect to $err(x)$) of the parameters $n_c$, $r \in [0.01, 10]$, $\lambda \in [0, 10]$ and $\Phi \in \{\Phi_1, \Phi_2, \Phi_3, \Phi_4\}$. As this new step introduces a second level of global optimization, it is only reserved to cases where the time evaluation of $x \mapsto J(x)$ is many orders of magnitude higher than the time evaluation of $x \mapsto \tilde{J}(x)$, as in a car drag reduction problem.

## 3 Comparison of Global Optimization Methods on Two Analytic Cases

Before applying them on a real drag reduction problem, all the previous global optimization algorithms have been tested and compared on two analytic test functions. These functions have been constructed in order to exhibit two behaviors that are supposed to be representative of a large number of realistic optimization problems. The first one is a Rastrigin type function with 3 parameters, directly inspired from the original one with 2 arguments:

$$Rast(x) = \sum_{i=1}^{3} \left(x_i^2 - \cos(18x_i)\right) + 3$$

defined on $\mathcal{O} = [-2,2]^3$ for which there exists many local minima (more than a hundred) and only a global minimum located at $x_m = (0,0,0)$ equal to 0.

The second test function is a modification of the classical Griewank function with 30 parameters:

$$Griew(x) = \sum_{i=1}^{30} \frac{x_i^2}{50} - \prod_{i=1}^{30} \cos(\frac{x_i}{\sqrt{i}})$$

defined on $\mathcal{O} = [-10,10]^{30}$ which has also a unique global minima at the origin but only few local minima. In order to achieve a quasi-certain convergence with a simple genetic algorithm, the population number and the maximal generations number are respectively fixed to $(N_p, N_g) = (30, 300)$ for the Rastrigin function and $(N_p, N_g) = (100, 160)$ for the Griewank function. The crossover and mutation probability have been set to their best observed value in this case, that is $p_c = 0.3$ and $p_m = 0.9$ (see [11] for more details).

Figure 1 displayed below gives an example of convergence history for the Rastrigin and Griewank function respectively, with six different global optimization methods that have been previously presented in paragraph 2 (GA, $HM_0$, $HM_2$, $HM_3$, $HM_4$ and AGA). Note that each gradient computation is counted as $2n$ function evaluations where $n$ is the dimension number of the search domain (3 or 30 here) and that each approximated evaluation of the function in the AGA algorithm is not counted as an evaluation. Due to the large number of curves displayed, the authors must apologize for the poor visibility on a black and white copy.

On this figure, it can be seen that each method that has been constructed gives better results than a simple GA on two different aspects: the first one is the computational time which is assumed to be directly related to the evaluation number of the function to minimize and the second one is the accuracy level that is reached at the end of the computation.

In order to give more quantitative results, a statistical study based on a set of 100 independent optimization processes has been realized. Table 1 gives the approximated average gain compared to a classical genetic algorithm (ie the evaluation number reduction rate for a given convergence level) that has been observed for three global optimization methods. Note that the results obtained

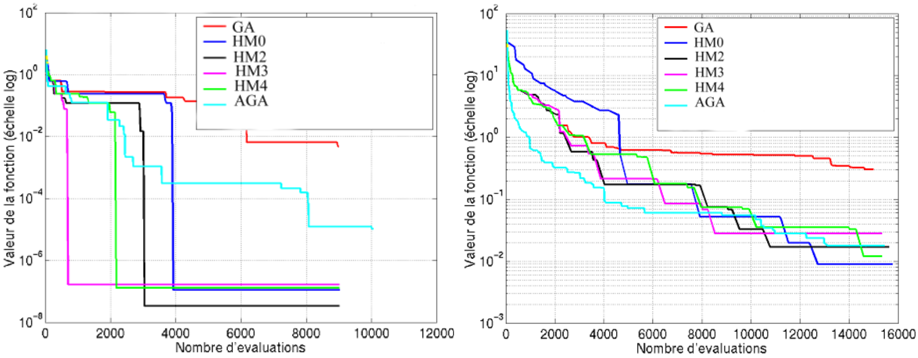

**Fig. 1.** Example of convergence history for the Rastrigin function (left) and the Griewank function (right) with 6 different global optimization methods

**Table 1.** Evaluation number reduction rate for $HM_0$, $HM_3$ and AGA compared to a simple GA for each test function

| | Rastrigin function | Griewank function |
|---|---|---|
| $HM_0$ | 2 | 2 |
| $HM_3$ | 10 | 4 |
| AGA | 4 | 10 |

for $HM_1$, $HM_2$ and $HM_4$ have not been displayed as they were not as good as those of $HM_0$ and $HM_3$.

This table confirms the interest of all the global optimization methods previously presented, but on a various degree. For example, it can be observed that $HM_3$ give better results than $HM_0$, which could not have been easily forecast before these tests. Thus, $HM_3$ and AGA seem to be the most promising ones in terms of convergence time reduction. One one hand, $HM_3$ appears to be more efficient than AGA on the Rastrigin function as the latter exhibits many local minima that can be rapidly tracked by the local search process. On the other hand, as the number of parameters increases, the AGA method performs better than the $HM_3$ method as it does not need any gradient computation which becomes very costly by finite differencies.

In view of their promising results, these global optimization methods are now used and compared in the next paragraph in the context of a car drag reduction problem.

## 4 Comparison of Global Optimization Methods for Car Drag Reduction

A classical 3D car drag reduction problem ([12]), already investigated in [1], has been extensively studied with the global optimization methods presented in

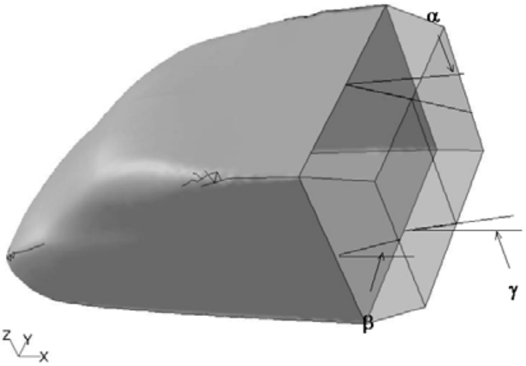

**Fig. 2.** 3D car shape parametrized by its three rear angles $\alpha$, $\beta$ and $\gamma$

paragraph 2. It consists in minimizing the drag coefficient (also called $C_d$) of a simplified car shape with respect to the three geometrical angles defining its rear shape (see Figure 2): the back light angle (called $\alpha$), the boat-tail angle ($\beta$) and the ramp angle ($\gamma$).

The drag coefficient to be minimized is defined by the following expression:

$$C_d \equiv C_d(\alpha, \beta, \gamma) = \frac{F_x}{\frac{1}{2}\rho V_\infty^2 S}$$

where $\rho$ is the mass density, $S$ the front surface, $V_\infty$ the freestream velocity and $F_x$ the longitudinal component of the aerodynamic force exerted on the car. The latter is obtained after a 3D turbulent Navier Stokes computation around the car. This computation, very costly and sensitive to the car geometry, explains the major difficulty of such optimization problem.

In order to improve the first optimization attempt presented in [1], the search domain has been reduced: here, $(\alpha, \beta, \gamma) \in [15, 25] \times [5, 15] \times [15, 25]$ (degrees) and the aerodynamic computation is done with a finer grid, namely with a 6 million cells mesh. In this context, one $C_d$ evaluation, done with a commercial CFD code, takes 14 hours CPU time on a single processor machine.

Three different types of global optimization methods have been compared on this problem. The first and reference type is a classical GA with a population number $N_p$ equal to 20 and $(p_c, p_m) = (0.3, 0.9)$. The second type consists of hybrid methods, HM$_0$ and HM$_3$, as described in paragraph 2.2 with gradient evaluations computed by centered finite differences. The third type is a GA with fast and approximated evaluations (called AGA) with or without a secondary optimization of the interpolation parameters $n_c$, $\lambda$, $r$ and $\Phi$: see paragraph 2.3.

The convergence history of all these optimization methods for the present drag reduction problem is depicted in Figure 3. In order to achieve a reasonable computational time, parallel evaluations on a cluster of workstations have been done.

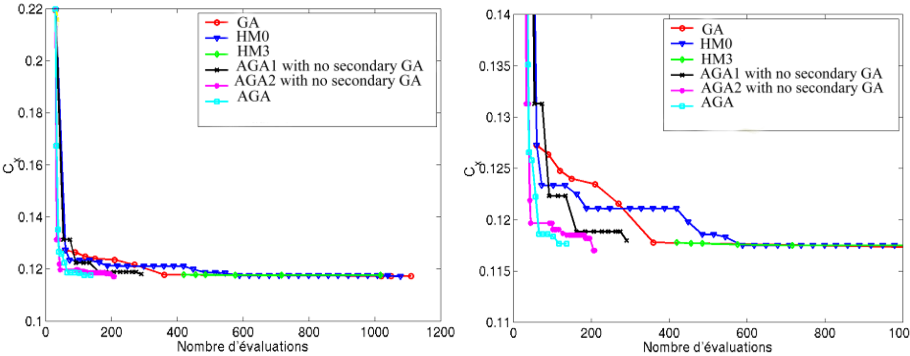

**Fig. 3.** Convergence history for drag reduction of a 3d car shape with six different global optimization methods (right figure: zoom of left figure)

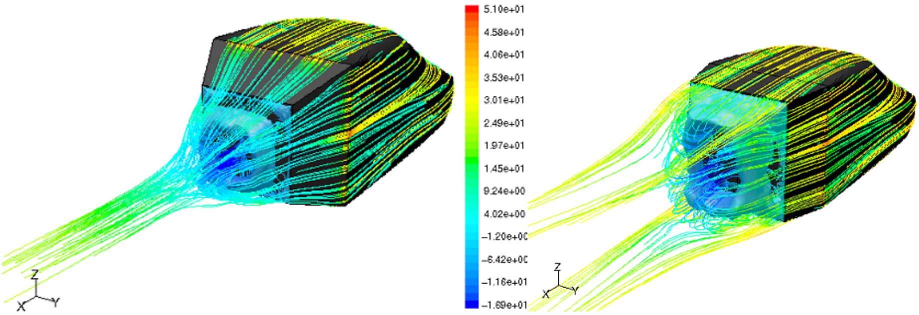

**Fig. 4.** Iso pressure streamlines coloured by the longitudinal speed for the lowest drag (left) and a high drag (right) car shape

This figure show in particular that all the methods have nearly reached the same drag value, around 0.117, starting with an almost double value, but with a different number of cost function evaluations. More precisely, the AGA algorithm has permitted to reduce this number by a factor 7 compared to a classical GA. Note that the AGA methods with a fixed set of parameters (called AGA1 and AGA2) exhibit a lower gain, which seems to justify the interest of a secondary optimization of the approximation parameters. On the other hand, the hybrid method $\text{HM}_0$ and $\text{HM}_3$ don't exhibit a significant improvement compared to the GA, likely because of the lack of accuracy in the gradient computation.

The optimal angles obtained by all the global optimization methods are nearly equal to $(\alpha, \beta, \gamma) = (17.7, 10, 18.4)$. These values have been experimentally confirmed to be associated with the lowest drag value that can be reached. The associated computational aerodynamic wake flow is depicted in Figure 4 and compared to an example of a high drag shape ($Cd = 0.22$).

It can be seen in particular that the optimized shape exhibits a narrow and regular recirculation volume behind the vehicle as predicted by many authors.

Note that such aerodynamic interpretation must be done on a 3D level and not from a longitudinal projection because of the real 'three dimensionality' of the flow.

## 5 Conclusion

In this article, two types of global optimization methods have been compared with the classical genetic algorithm method (GA). The first type, called hybrid methods (HM), consists in including a local search process for some well chosen individuals during the GA evolution. The second type, called AGA, incorporates a fast but approximated evaluation process for some individuals. For all the tested problems, the needed number of cost function evaluations to achieve global convergence has been largely decreased, between a factor 2 and 10 compared to a classical GA. For instance, in a classical car drag optimization problem, the AGA method with a new surrogate model has permitted to obtain the lowest drag car shape 7 times quicker than any other existing method. Such promising method is now ready to be applied to reduce the aerodynamic drag coefficient of more and more realistic car shapes with a larger number of parameters.

## References

1. Muyl F., Dumas L. and Herbert V.: Hybrid method for aerodynamic shape optimization in automotive industry. Computers and Fluids, **33** (2004) 849-858
2. Goldberg D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)
3. Poloni C.: Hybrid GA for multi objective aerodynamic shape optimization. Genetic algorithms in engineering and computer science. John Wiley and Sons, **33** (1995) 397-415
4. Renders J.M. and Flasse S.P.: Hybrid methods using genetic algorithms for global optimization. IEEE Transactions on systems, man and cybernetics, **26** (1996) 243-258
5. Vicini A. and Quagliarella D.: Airfoil and wing design through hybrid optimization strategies. AIAA paper (1998).
6. Ong Y.S., Nair P.B., Keane A.J. and Wong K. W.: Surrogate-Assisted Evolutionary Optimization Frameworks for High-Fidelity Engineering Design Problems. Knowledge Incorporation in Evolutionary Computation. Studies in Fuzziness and Soft Computing Series, Springer Verlag (2004) 307-331
7. Jin Y.: A survey on fitness approximation in evolutionary computation. Journal of Soft Computing, **9** (2005) 3-12
8. Jin Y., Olhofer M. and Sendhoff B.: A framework for evolutionary optimization with approximate fitness functions. IEEE Transactions on Evolutionary Computation, **6** (2002) 481-494
9. Ong Y.S., Nair P.B. and Keane A.J., Evolutionary Optimization of Computationally Expensive Problems via Surrogate Modeling. AIAA Journal, **41** (2003) 687-696

10. Giannakoglou K.C.: Acceleration of GA using neural networks, theoretical background. GA for optimization in aeronautics and turbomachinery, VKI Lecture Series, (2000)
11. Muyl F.: Méthode d'optimisation hybrides: application à l'optimisation de formes arodynamiques automobiles. Phd thesis Université Paris 6 (2003)
12. Sagi C.J., Han T., Hammond D.C.: Optimization of bluff body for minimum drag in ground proximity. AIAA paper (1992)