8th World Congress on Structural and Multidisciplinary Optimization
June 1 - 5, 2009, Lisbon, Portugal

Robust shape optimization in aeronautics

Jordi Pons-Prats[1], Gabriel Bugeda[2], Eugenio Oñate[3], Francisco Zárate[4], Jorge Hurtado[5]

[1] CIMNE – UPC, International Center for Numerical Methods in engineering, Technical University of Catalonia, Barcelona, Spain. jpons@cimne.upc.edu
[2] CIMNE – UPC, International Center for Numerical Methods in engineering, Technical University of Catalonia, Barcelona, Spain. bugeda@cimne.upc.edu
[3] CIMNE – UPC, International Center for Numerical Methods in engineering, Technical University of Catalonia, Barcelona, Spain. onate@cimne.upc.edu
[4] CIMNE – UPC, International Center for Numerical Methods in engineering, Technical University of Catalonia, Barcelona, Spain. zarate@cimne.upc.edu
[5] Department of Civil Engineering, National University of Colombia, Manizales, Colombia. Jhurtado14@une.net.co

1. Abstract
Optimization is becoming an important field of research. The availability of more powerful computational resources, the ever-seek better performance and the company needs regarding time-to-market reduction, leaded to the necessity of better designs in less time.
Aeronautics field has not been an exemption; shape optimization is a largely studied problem. It can be applied to many disciplines in this field, always intended to improve existing performances, and reduce costs and annoying characteristics like noise or consumption.
Traditionally, optimization procedures were based on deterministic methodologies [6] so, it means that optimization leads to an optimum value which is an optimal point. Engineers have realized that this optimal point is no longer their aim. They need to ensure the optimal behaviour in the whole operating range; if they do not consider what happens in the vicinity of this point, a problem can arise. That is, in many cases, if the working point differs from the original, even a little distance, efficiency is reduced considerably [9].
Non deterministic methodologies have been applied to many fields [7][8], but the time consuming calculations required on CFD avoid to be applied to shape optimization. The study of the variability of the result against variability of the input parameters is a better representation of the real world; using these kinds of methodologies we can simulate from manufacturing tolerances to unknown measurement errors. The result is no longer a point, but a range or a set of values which defines the area where, in average, optimal output values are obtained. The optimal value could be worse than other optima, but considering its vicinity, it is clearly the best.
This communication presents the development of a non-deterministic strategy coupled with optimization techniques. The main aim is the definition of an optimization procedure that ensure greater robustness of the solution than usual procedures. In order to reach this aim, probabilistic data will be used to define input parameters, and applied to each one of the members defined by the optimizer. In our case, we will use a genetic algorithm as optimizer.
2. Keywords: Robust design, Non-deterministic optimization, Aeronautics.

3. Introduction
Many engineering applications require solving optimization problems to define the optimal design. In many cases, optimization problems include nonlinear objective functions, and/or nonlinear constraint functions. But, even if these functions were linear, engineering optimization problems might be too complex to be solved easily (like in the fluid dynamics case) because they cannot be solved analytically or might present local solutions. Traditional approaches usually converge to these local optimal solutions [1][2], because their search range is limited to a neighbourhood of the starting point.
Uncertainty is also an important parameter to be taken into consideration. Management of uncertainty over the simulation process will produce better results regarding the robustness of the design. Robustness means that result values are not greatly modified by little variations of the input variables.
Optimization methods like Genetic Algorithms, which are considered as probabilistic ones due to their probabilistic generation of generation members, only consider probabilistic information of the actual input values. Uncertainty applied to boundary conditions are not considered, even cannot be controlled using the input values of the genetic algorithms.
In our case we try to avoid all problems derived from local solutions and uncertainty. The probabilistic-stochastic method developed here, like the general stochastic optimization methods, has the ability to escape from local solutions, by analyzing the whole search space, and applying an environmental uncertainty under the profile

analyzed. Main input variables, id est. geometry definition parameters of the airfoil, are controlled by genetic algorithms. Secondary input parameters, id est. boundary conditions like angle of attack and air speed, will be defined using additional probabilistic data.

3.1. State of the art

Optimization methods using stochastic and non-deterministic procedures have not been largely applied to aerodynamic shape optimization. It is because of the large amount of calculations needed.

Huyse and Lewis [9] focused on the development of an effective strategy based on stochastic optimization. They define a non-deterministic optimization procedure which achieved the best performance and minimal cost, managing the uncertainty, and obtain the best multi-point solution. They use the Von Neumann – Morgenstern decision theory to decide the best optimization criterion. The result is analyzed evaluating the risk of the solution. Other works have studied the robust optimization method. Nagarathinam et al [10] had clearly differentiated between non-deterministic and robust optimization methods. The evolutionary algorithm is improved using the so called Hierarchical Genetic Algorithm, which uses a hierarchical topology of the members of the population. These are distributed into three layers, one used by the refined calculations, one by the coarse calculations and exploration for new optimal points, and one intermediate layer which is used as "bridge layer" between refinement and exploration layers. Several other works studied deterministic procedures such as the one by Obayashi et al [12], and Mitra [19], where both defined deterministic optimization methods using Evolutionary or Genetic Algorithms. Both of them are clear examples of the difficulty of this kind of applications, due to the complexity of the fluid dynamics analysis. Simplification is a commonly used tool. Analyses are restricted to a value of Mach number, or Reynolds number for simplification purposes.

3.2. Genetic Algorithms

The so called evolutionary algorithms, and genetic algorithms in particular, are methods based on the calculation of a population of solutions which evolves to the best solution. The basic steps of a solution are as follow:

1. Initialise a population of values.
2. Evaluate each member in the population.
3. Create new values, combining or mating the current values.
4. Delete members of the population which are no longer used.
5. Evaluate new values and insert them to the population.
6. Iterate from step 3 while the defined criteria are not accomplished.
7. If criteria are reached, stop calculations and return the fittest values.

Those are the descriptive steps of an evolutionary optimization method. Each one generates, mutate and mate the members using different procedures. Some of them will generate only new members from the information obtained of the previous population. But some others, the so called stochastic programming techniques, generate an entire population from the information of the fittest members of the previous population.

Based on nature techniques of reproduction, Genetic algorithms use crossover, mutation and selection to create new members of the next generation. Also the technique of elitism can be applied to increase the converge ratio.

In this work we began to use a single-objective genetic algorithm from Prof. K. Deb, but finally we decided to move to a more evoluted Genetic Algorithm code developed by Deb [14]. The code is based on the so called Non-dominated Sorting Genetic Algorithm II which reduces the computational complexity and introduces elitism. Genetic algorithms are extensively used on the optimization of the aerodynamic shape of profiles [10][11][19][12][20], even wing shapes [21]. Generally speaking, genetic algorithms are used in conjunction with CFD solver and are described as probabilistic optimization method due to the definition of new generations in each step. The probabilistic behaviour of the method has no relation with the statistical definition of the input variables or the statistical analysis of the output variables, and then they cannot be considered non-deterministic methods.

3.4. Non-deterministic optimization

Both, deterministic and non-deterministic optimization method are developed in many fields, but, due to the complexity and the required long calculation time computer fluid dynamics (CFD) needs, non-deterministic optimization was not applied to this field so far.

The so-called deterministic optimization does not use any information about input variables but range and search space. Mathematically, it means that all values are considered as equally probable. But engineering problems do not have such behaviour and, for instance, some variables could follow a normal density probability function.

Taking this behaviour into consideration requires more calculation time due to the generation of more members and populations. Although it is applied in other fields, CFD requires a long time calculation for each member, so the total computational cost is rapidly increased. This is the main reason why non-deterministic optimization was not applied to CFD so far. Marco [11] had found similar problems with the time required by the CFD calculations, and used the Euler equations to calculate the CFD results and Genetic Algorithms as the optimization method. Jeong et al [13] used a Kringing Model in order to reduce the amount of time required on the optimization method. Kringing Model or response surface model are similar to Neural Networks in that they use an approximation model to avoid the real calculations. Plevris et al. [8] define a statistical method combining Monte Carlo simulation and Latin Hypercube Sampling design. They compare the results from a deterministic formulation and a robust design formulation, understanding robust as the less sensitive to uncertainties.

Non-deterministic or robust analysis is based on sampling methods. These methods produce a number of sampling points that must follow a given statistical distribution. The search space can be modelled using the Montecarlo technique, but also Latin Hypercube sampling could be used. Both sampling techniques can accurately model the probabilistic behaviour of a value. It has been shown that Latin Hypercube Sampling is able to model the random behaviour with a fewer number of samples. Table 1 tabulates comparative values between the Montecarlo technique, used as reference, and Latin Hypercube Sampling (LHS) technique. We defined a test case based on the calculation of Cl/Cd ratio of an airfoil, so we will compare the mean and standard deviation values obtained using Montecarlo and LHS samplings. 250-Montecarlo-samples analysis is used as reference, which is compared with LHS analysis using from 25 to 250 samples. 25 LHS-samples analysis has a deviation of 0.05% from 250 Montecarlo-shot analyses. The confidence interval defined using the +/-3σ range has differences lower than 0.15%. These values confirm the ability of LHS against Montecarlo technique in order to reduce the number of samples required for approximate the search space, fact that can be translated to a reduction of the calculation time required by the non-deterministic analysis.

Table 1: Comparison between Latin Hypercube Sampling (LHS) and MonteCarlo Sampling (MC)

	Mean Cl/Cd	Standard deviation Cl/Cd	*Confidence range*		*Mean Deviation MC vsLHS*	-3σ value *Deviation MC vs LHS*	3σ value *Deviation MC vs LHS*
			-3σ	3σ			
25 samples using LHS	23,52	0,073	23,30	23,74	-0,050%	0,028%	-0,127%
50 samples using LHS	23,51	0,079	23,28	23,74	-0,065%	-0,065%	0,065%
100 samples using LHS	23,53	0,078	23,30	23,76	0,002%	0,018%	-0,015%
150 samples using LHS	23,53	0,077	23,30	23,76	0,009%	0,027%	-0,010%
200 samples using LHS	23,53	0,078	23,30	23,77	0,011%	0,022%	0,001%
250 samples using LHS	23,53	0,079	23,29	23,77	0,000%	0,000%	0,000%
250 samples using MC	23,53	0,079	23,29	23,77			

4. Procedure and methodology

The first step to define a non-deterministic optimization procedure is to perform a stochastic finite element analysis of each design. In this work, two available codes are integrated to work together; STAC [16], a stochastic analysis management tool, and a CFD solver, which can be TDYN[17], an incompressible flow solver, or PUMI[22], a compressible flow solver.

STAC is a powerful tool which enables to define several types of probability density functions applied to input variables of the solver (TDYN or PUMI), and produce a number of shots in order to obtain a set of sampling points with the given statistical distribution. STAC is able to use both Montecarlo sampling and Latin Hypercube sampling.

STAC is a Windows-developed software which provides control of any kind of solver from the command line. If the solver can be executed using the command line, STAC has the ability to work using customized applications which launch the information from STAC to the solver defining the necessary input configuration. Similarly, STAC can pick up the results from the solver in order to analyse them. STAC can be seen as a pre and post-processor for stochastic analysis.

TDYN code solves Navier-Stokes equations using a stabilized finite element method. It is a Windows native application that has a user friendly environment and a graphical interface. PUMI is a high-efficiency compressible flow solver developed in CIMNE, and mainly intended to deal with large scale problems. Both TDYN and PUMI can be launched using the command line, so STAC uses this capability to modify the input file and launching the solver. In order to modify, as desired, the input files of the solver some customized codes have been necessarily developed.

Non-deterministic optimization has been mainly applied to aerodynamics and other fields using little variability of the input variables. Thanks that STAC provides the capability to define statistical behaviour all the search space wide, and the capability to manage the input and output information generated the procedure defined is based on the integration of both software. From STAC, the statistical information is assigned to each stochastic input variable, and then transferred to the solver via a customized code, as mentioned. Using stochastic input variables, plus additional ones, the solver is launched and a result can be obtained. This output value is transferred to STAC to be processed and statistically analyzed. Any kind of input variables can be defined; from mesh sizes, to material properties and geometric parameters. Generally speaking, all the information inputted to the solver can be controlled by STAC, but it is not necessary to define as stochastic variables all of them.

4.1. Output variability versus mesh size variability

The first test of the developed software package was intended to analyse the variability of the C_L/C_D ratio of an airfoil versus the variability of the mesh size. Considering the environment conditions of the analysis, a low Mach number, and a quite laminar flow, the results were a validation of the procedure defined. Three input mesh sizes were applied to the geometry, on both leading edge and trailing edge points of the 2D-profile, on both upper and lower profile lines, and finally on the control area surface. Both points use the same value, so only one parameter is considered. A single parameter for the lower and upper lines of the profile was also used. Finally, we considered a single parameter to the general mesh size applied to the control area. Figure 2 shows the mesh identifying the defined mesh sizes.

In order to analyse the variability effect on the result, two different types of probability density function were applied to each one of the input parameters. The combination of the groups of one, two or three parameters define the main steps, but also each applied probability density function adds new information.

As mentioned above two different probability density functions (PDF) were applied to the mesh sizes. Normal PDF was defined to centre the analysis on the mean value and the standard deviation was used to evaluate the effect of each input parameter on the output result, while a uniform PDF was used as a control calculation. Roughly, the uniform PDF was defined to spread random values in the same range than the normal PDF previously defined.

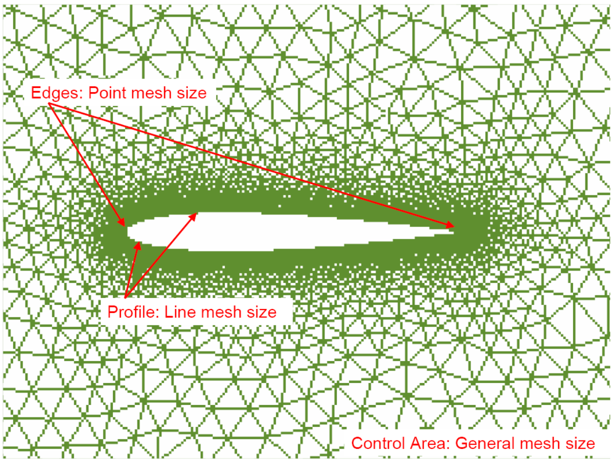

Figure 1.- Mesh sizes location

The main conclusion, as expected, points out that the mesh size and its variability, applied on both leading and trailing edges points of the 2D-profiles has the most important effect on the variability of the result. The second most important effect is produced by the mesh size applied to profile lines. The graph clearly shows large standard deviation range for those analysis where only variability on points is analyzed. If we focus our attention to the three analysis called Pt, Ln, and Gl, where single mesh size is analyzed each time, we can realize on the different effect of each size and its variability on the variability of the output value, the C_L/C_D ratio.

4.3. Output variability versus environmental variables variability

The developed integration of Montecarlo analysis management tool and CFD solver is now used to determine the effects of the variability of environmental parameters on the variability of the results. The environmental parameters chosen are the angle of attack, the velocity of the airflow (Mach number) and the turbulence of the flow (the Reynolds number). In order to facilitate the input variables definition, the Mach number and the Reynolds number are split into air velocity, air density and air viscosity. The nomenclature used is ‘a’ as angle of attack, ‘v’ as air velocity, ‘ds’ as air density, and ‘vs’ as air viscosity.

The study is conducted in a similar way as using mesh variability. Normal and uniform probability density functions are defined for each parameter and several combinations of them enables the analysis of the weight of each one on the variability of the C_L/C_D ratio. Figure 4 shows the mean values accompanied by the +/-3σ range, in a min-max graph representation. The nomenclature used to identify each analysis on the graph is as follows; “normal” means a normal PDF applied, “normxAA” means a normal PDF using standard deviation multiplied by AA value, and finally, “x” identifies the stochastic variables defined.

A number of analyses are performed for subsonic and laminar flows, with low Mach number and moderate Reynolds number. The variability of the angle of attack is the value which affects most the variability of the results, as expected. Considering that the analyzed result is the ratio of C_L and C_D, the density and viscosity of the air could be excluded from the analysis because their low effect on the C_L/C_D ratio. The conclusions of the case study coincide with those expected which served as a general validation of the method.

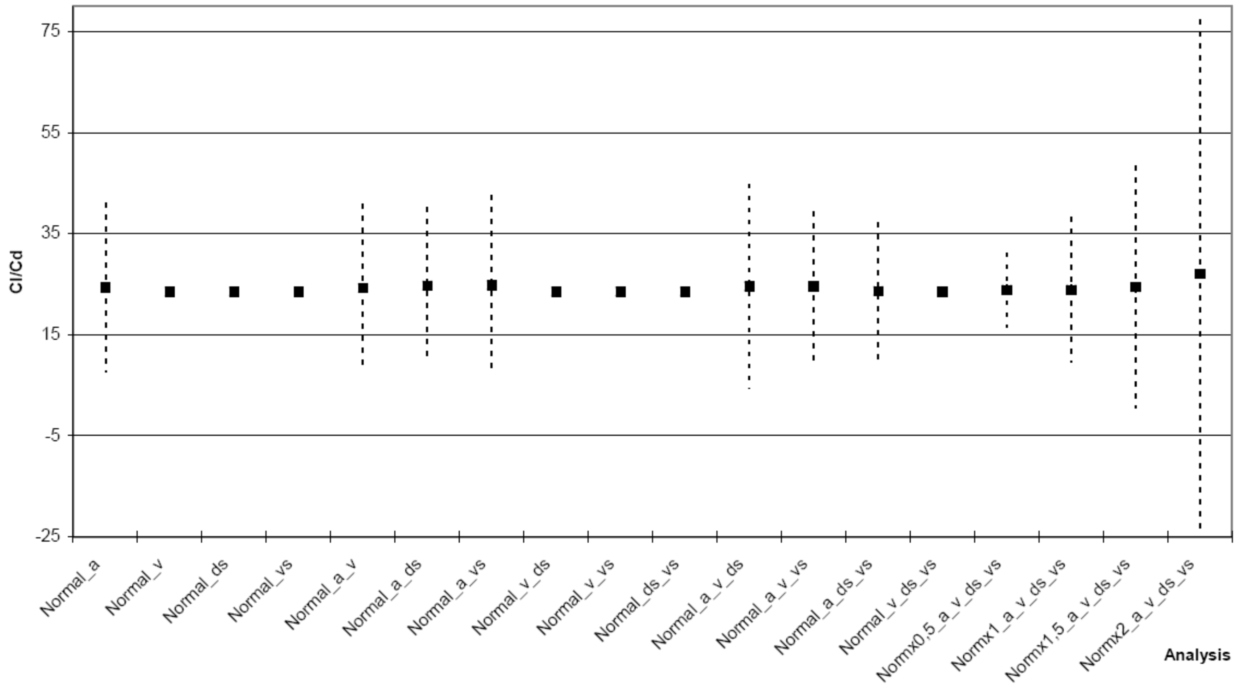

Figure 2.- Mean and +/-3 standard deviation range: Variability due to environmental variables

Similarly as done on the analysis of the mesh size, Figure 2 shows the large variability when variability on the angle of attack is defined, but negligible effect due to air density and air viscosity variabilities.
This result can not be extrapolated to the case of a compressible flow or with large turbulence, because viscosity variability will affects to drag, and density to lift in a completely different way as done in the analyzed case.

4.4. Non-deterministic optimization
The previous work built the basis for a non-deterministic optimization procedure. The integration of a stochastic analysis management tool as STAC and a computer fluid dynamic solver, as TDYN or PUMI, shows that a non-deterministic optimization method can be defined.
A new procedure for a non-deterministic optimization is presented. It uses the capacity developed on the previous stage of the research, and represents an improvement in the sense that variability is part of the study.
As a first approximation to the method, a well-proven optimization procedure is developed. Evolutionary algorithms have demonstrated their potential in order to reduce optimization steps, and to converge quickly and efficiently to the optimal value [18]. Their ability to seek in all the search space and to avoid local minima is their best qualities.
The main difference between deterministic procedures and the one developed here is the fact that the objective function comes from the mean and standard deviation values of a cloud of points generated using the Montecarlo method. Considering the optimization of the geometry of the airfoil as our target, regarding its lift and drag performance, each member of each population of the genetic algorithm is evaluated under a cloud of random environmental variables (namely, angle of attack, air velocity, density and viscosity). From the cloud of points obtained, we take the mean and the standard deviation to obtain the fitness of each member.
Environmental variables are defined using their mean values to point out the working point, but also using their standard deviation in order to capture their intrinsic behaviour; that is, measurement errors, or the natural variability of the parameter.
Some promising results are presented showing the potential of the optimization method. Due to the featuring of the Montecarlo analysis, it is however a time-consuming method and it takes a long time for each optimization step to

end. Few generations were calculated due to the total amount of time required (Figure 3). That fact produces a trend that cannot be clearly defined, although it can be considered as an increasing trend.

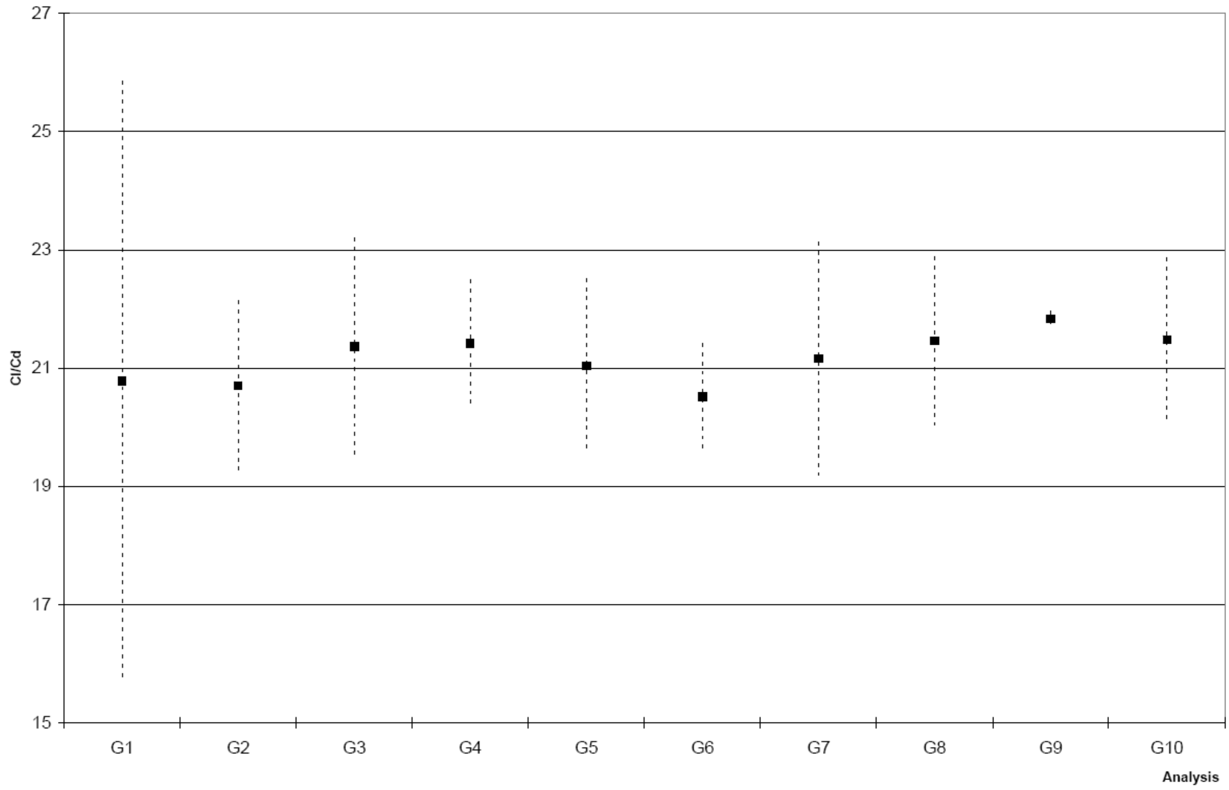

Figure 3.- Mean and +/-3 standard deviation range: Cl/Cd ratio evolution; mean and standard deviation

Several actions are considered for improving computational efficiency. The most significant one is the adjustment of the number of Montecarlo shots, however the larger the number of shots, the more precise the Montecarlo analysis, so this item implies a limitation on itself. As previously mentioned, using Montecarlo technique is time-demanding, and its cost in computational time is unaffordable in order to reach a good optimization procedure. The case presented is a clear example of the cost of calculating few generations. Because the problem is mainly the time required to calculate the CFD analysis, the solution leads to the need of modeling the results from the CFD solver. A Neural Network provides us this capability, and it is easily introduced into the Genetic algorithm code as described in the next section.

4.5. Non-deterministic optimization using Neural Networks

Neural Networks provide a powerful tool for reducing the calculation time. After a required training, the network will obtain a result much faster than performing the calculation itself. Based on the work of Lopez [4], we define an embedded Neural Network into the genetic algorithm code. We consider as Neural Network, a Multilayer Perceptron Model [5][15].

The non-deterministic optimization procedure is now calculated using the intelligence of a Neural Network, which we educate to provide us with the solver output. The time required to obtain each optimization steps is drastically reduced, so it becomes competitive against standard optimization methods. Required time is reduced from 48h to few minutes. The method guidelines are described on figure 4.

Initial Population

Calculation of the fitness of each member – using neural networks

Genetic Algorithm main loop

Generate new Population

Selection, Cross-Over and Mutation – based on GA techniques

Calculation of the fitness of each member – using neural networks

End of iterations

Convergence achieved

Maximum number of generations reached

Figure 4.- Genetic Algorithm + Neural Networks structure

The main difference between the non-deterministic procedure previously defined and this one is the amount of information we obtain from the FEM analysis that is avoided using the neural network. The neural network is trained to provide C_L and C_D coefficients, which will be used by the objective function evaluation in GA code, in order to calculate the mean and the standard deviation if required. But from FEM analysis we can also study the lift, the drag, and C_L, and C_D of individual shots, which in some cases could help to understand the final result.
After several first test cases using single-objective genetic algorithm, we decided to move to NSGA-II code. NSGA-II is a multi-objective[14]

5. Results and conclusions

Problem to be solved is basically the optimization of a 2D profile. Coordinates of the knot points which define the upper and lower profiles are the input values of the genetic algorithm, in both single and multi-objective problems. In addition, we defined what we called the environment conditions as secondary input parameters. These parameters are angle of attack and Mach number, which are defined using a probabilistic definition. We applied a probabilistic distribution to each one so we obtain a set of samples to be applied to each genetic algorithm population member. So we can ensure the robustness of the solution against the input variability.

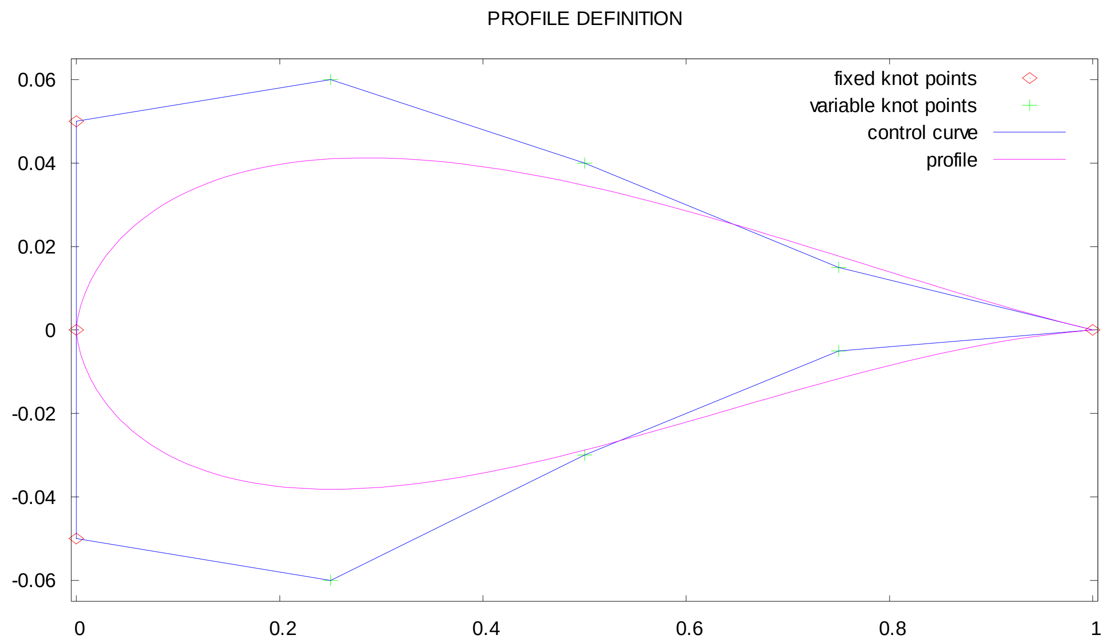

Figure 5.- Genetic Algorithm + Neural Networks structure

Knot points define the profile as seen in figure 5. In order to ensure the profile shape, some of them are considered as fixed, and some of them are variable, so it means they are the input variables of the GA. For Angle of attack and

Mach number values Normal Probabilistic distributions were defined, in order to capture their uncertainty, but seeking the objective values of their means. Table 2 shows the used values.

Table 2: Problem definition values

Name of the variable value	*Knot Points*		*Value ranges*		*Type of distribution*
	X coordinate	*Y coordinate*	*Lower Limit*	*Upper limit*	
Knot coordinates x1s, y1s	0	0	-	-	-
Knot coordinates x2s, y2s	0	0,05	-	-	-
Knot coordinates x3s, y3s	0,25	Variable	0,05	0,085	Random
Knot coordinates x4s, y4s	0,5	Variable	0,03	0,06	Random
Knot coordinates x5s, y5s	0,75	Variable	0,01	0,02	Random
Knot coordinates x6s, y6s	1	0	-	-	-
Knot coordinates x2l, y2l	0	-0,05	-	-	-
Knot coordinates x3l, y3l	0,25	Variable	-0,06	-0,03	Random
Knot coordinates x4l, y4l	0,5	Variable	-0,035	-0,02	Random
Knot coordinates x5l, y5l	0,75	Variable	-0,015	-0,005	Random
Probabilistic data					
	Mean	*Standard Deviation*	*Value ranges*		*Type of distribution*
			Lower Limit	*Upper limit*	
Angle of attack	4	0,5	2,5	5,5	Normal
Mack number	0,7	0,08	0,46	0,94	Normal

Several tests were performed during the validation process. First we used a single objective Genetic Algorithms from Prof. K. Deb. Two other initial conditions that we test are the spread of the initial population and the number of members in. Finally we decide to use a random definition of the initial values in other to avoid the following effect.

The most significant effect is related to the distribution of the initial population. Reaching quite the same optimum value, a clear convergence to the optimum occurs if the population is located in a certain area of the search space. The first graph, Figure 5 shows the irregular trend of the optimum values during the analysis. Despite its lack of tendency, all the values are quite similar and are within a range of 0,05% of the optimum value.

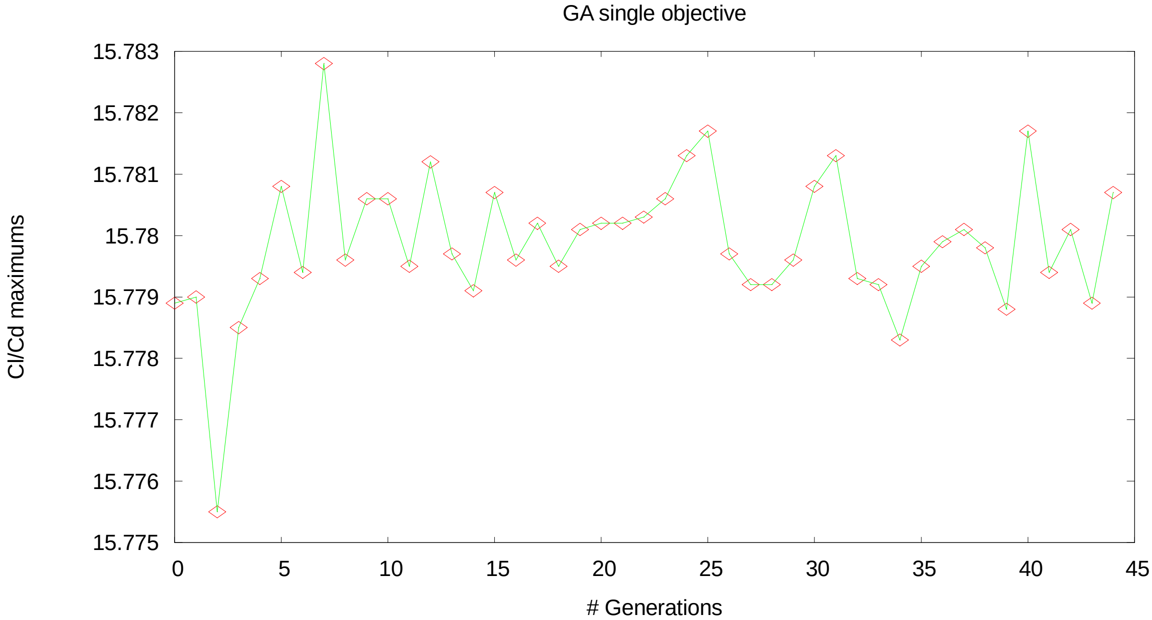

Figure 6.-Cl/Cd maximums evolution in single objective optimization; regular distributed initial population

If we check what happens with the means values of Cl/Cd ratio it shows a quite constant evolution, with some generations that lost some quality (decreasing its mean value because of the intrinsic GA search mechanism).

Figure 7.-Cl/Cd member means evolution in single objective optimization

On the contrary, when the initial population is located close to a certain area, the behaviour of the analysis is completely different. The evolution of the maximum values converges to the optimum in an increasing way, reaching quite the same optimum value as in the previous case. Both analyses can be said to converge to same optimum value. Figure 7 shows the maximum value evolution of each calculated generation.

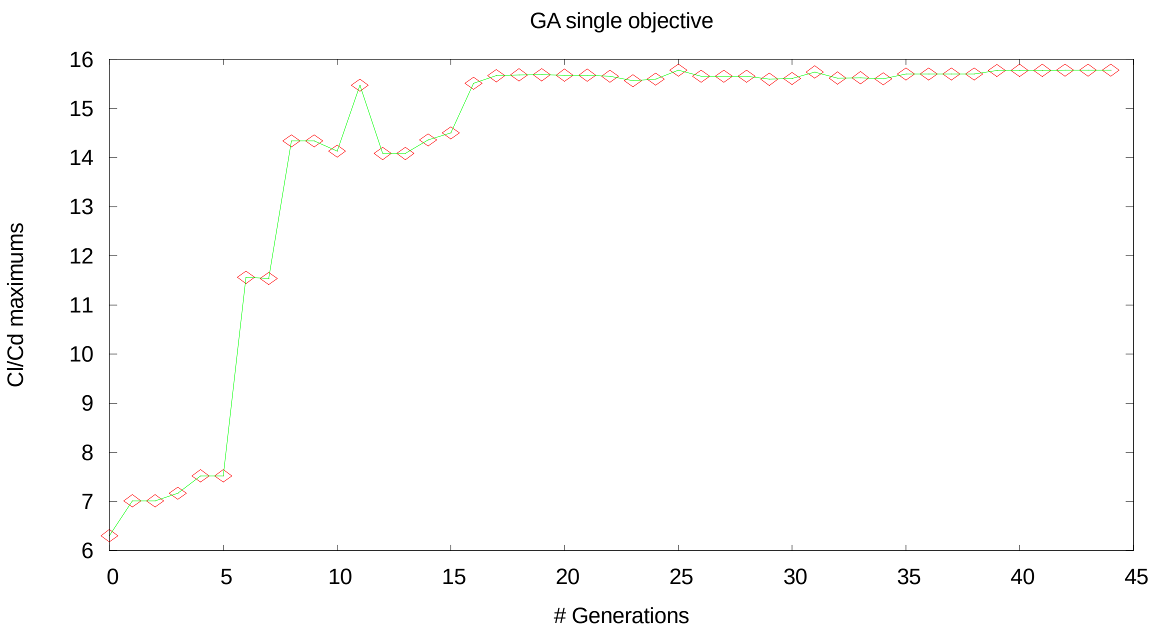

Figure 8.- Cl/Cd maximums evolution in single objective optimization; concentrated initial population

Figure 8 shows the evolution of the Cl/Cd means of each generation. We understand that mean values are most representative of the generation.

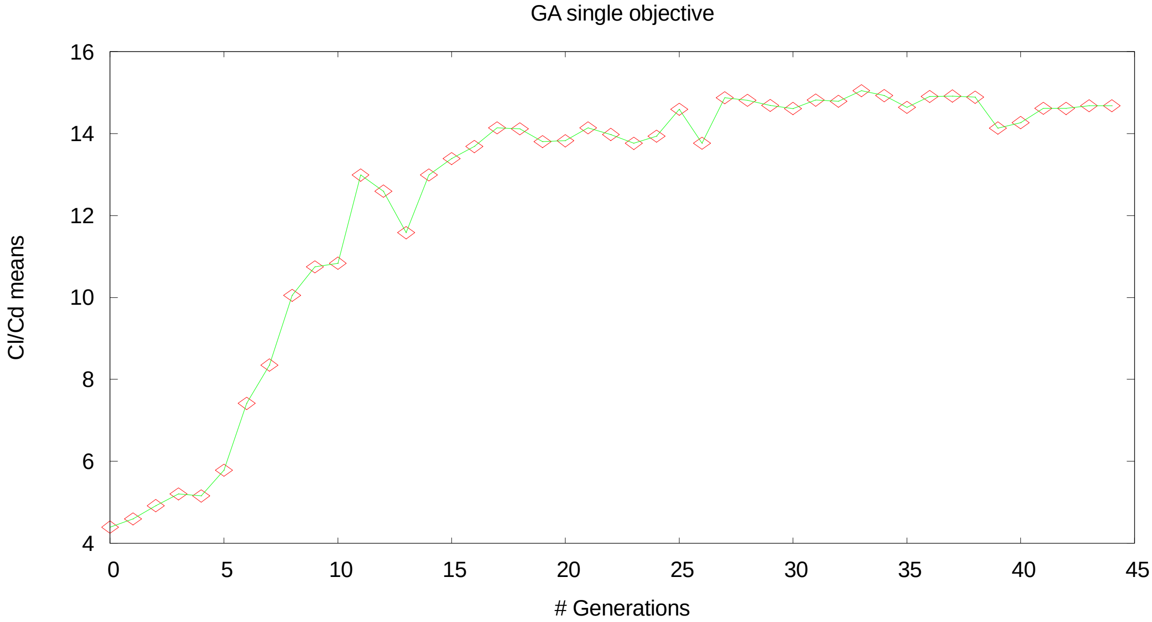

Figure 9.- Cl/Cd member means evolution in single objective optimization

Of course, genetic algorithm demonstrates its whole power solving multi-objective problems. As mentioned before, we decided to use NSGA-II from Prof. Deb, which provided us multi-objective capability and state-of-the-art algorithm.
We defined two objective functions to minimize; the first one is inverse of Cl (because we really want to maximization in this case), and the second one Cd. We compared the obtained results from the simple genetic algorithms coupled with neural network with non-deterministic defined analyses shown in figures 10 and 11. Figure 9 shown the whole population obtained from genetic algorithm coupled with neural network, which was previously trained using compressible flow data..

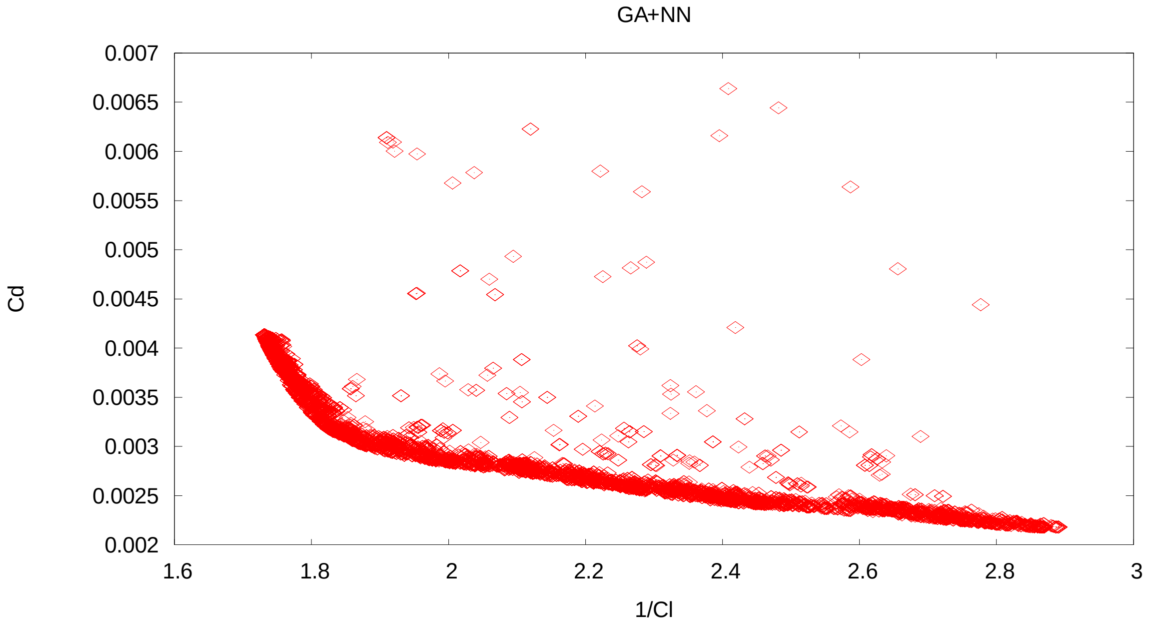

Figure 10.- Pareto Front and whole results; Genetic algorithm coupled with Neural Network

The first non-deterministic analysis used a MonteCarlo sampling technique in order to define stochastic data to be applied to angle of attack and Mach number. See figure 10 for the multi-objective optimization output data. Input variables to GA are the Y coordinates of the knot points that define the upper and lower profiles.

Several tests were performed in order to evaluate the most appropriate Montecarlo sampling to be used. Amount of samples from 10 to 250 of probabilistic data were used. Results obtained using only 10 samples had to be discarded because they did not accurately represent the search space of the input data. Increasing the amount of samples to 50, an increased accuracy of the search space was obtained; which leads to a better result. The selected number of samples was 250 in order to ensure the best probabilistic representation of the input data search space, and because it did not mean a major increment of computational cost.

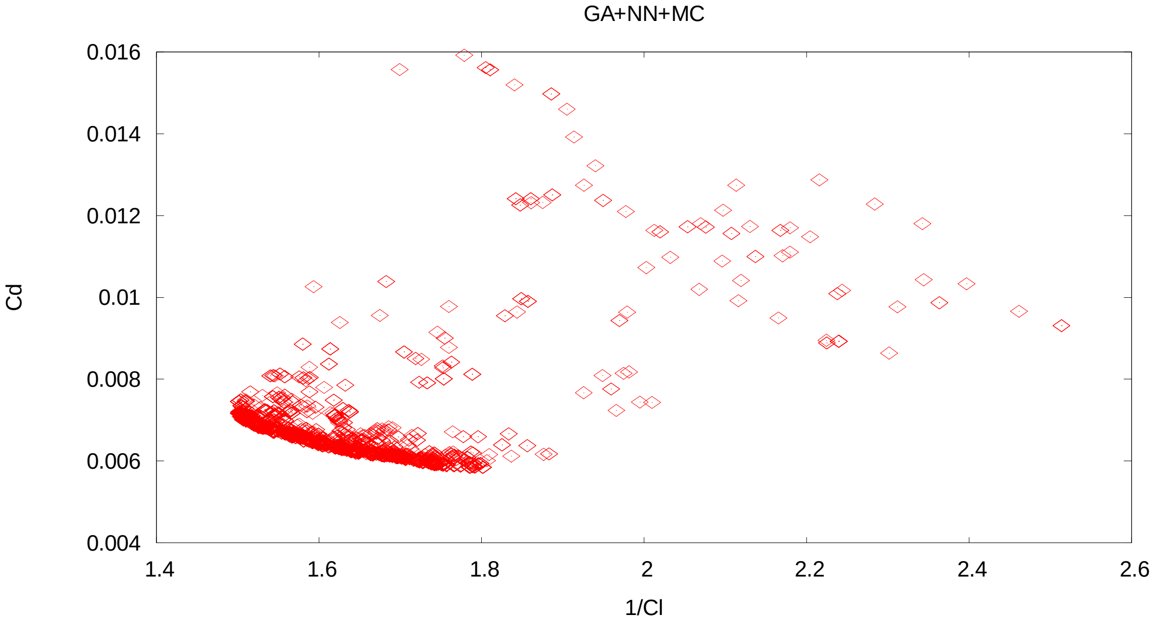

Figure 11.- Pareto Front and whole results; Genetic algorithm coupled with Neural Network and Montecarlo sampling

The second non-deterministic analysis used a Latin Hypercube sampling technique in order to define those data to be applied to angle of attack and Mach number. See figure 11 for optimization results. Profile definition, and genetic algorithm definition in these three analyses are the same as the previous ones.
Latin Hypercube sampling used 10 samples to model the input value space. If we compare this amount of samples with the required one when using Montecarlo sampling, we can realize the computational cost saving we can obtain with LHS. In addition, if we compare both Pareto fronts, we can realize that LHS provides a better representation of the whole input value space, which leads to shorter fronts and better solutions than using MonteCarlo sampling.

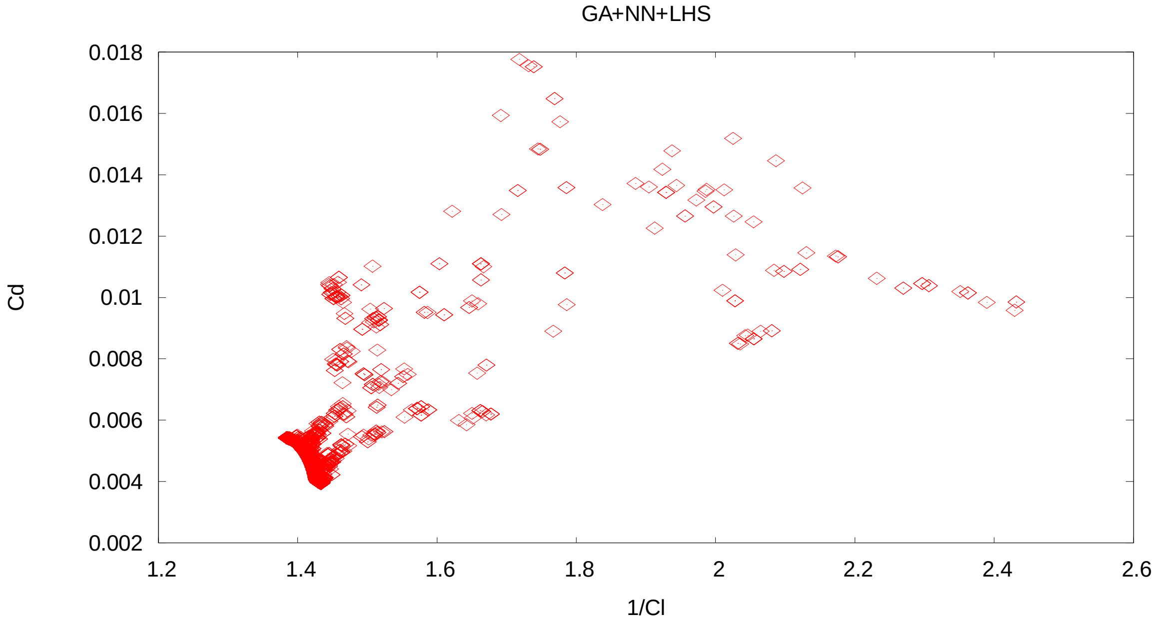

Figure 12.- Pareto Front and whole results; Genetic algorithm coupled with Neural Network and Latin Hypercube sampling

Comparing the three obtained Pareto fronts, in figure 12, those defined using probabilistic data shows a greater convergence. Both of them have a short front length compared with the non-probabilistic front. It means that those methods including probabilistic information are able to select the best values, reducing the range, and distinguishing between the most robust optima.
We can also compare Pareto front obtained using MonteCarlo and Latin Hypercube sampling. Latin Hypercube sampling produce shorter front because the sampling technique is able to better represent the search space. MonteCarlo sampling produces a more scattered plot and a long Pareto front. Even though sampling techniques are working on and defining the same search space, each one of them are creating different values set. It can justify the different values shown in the graph. Considering this fact, and the better representation of the search space using Latin Hypercube, we can agree that Latin Hypercube sampling, used to define probabilistic data for genetic algorithm and neural networks, obtains better results.

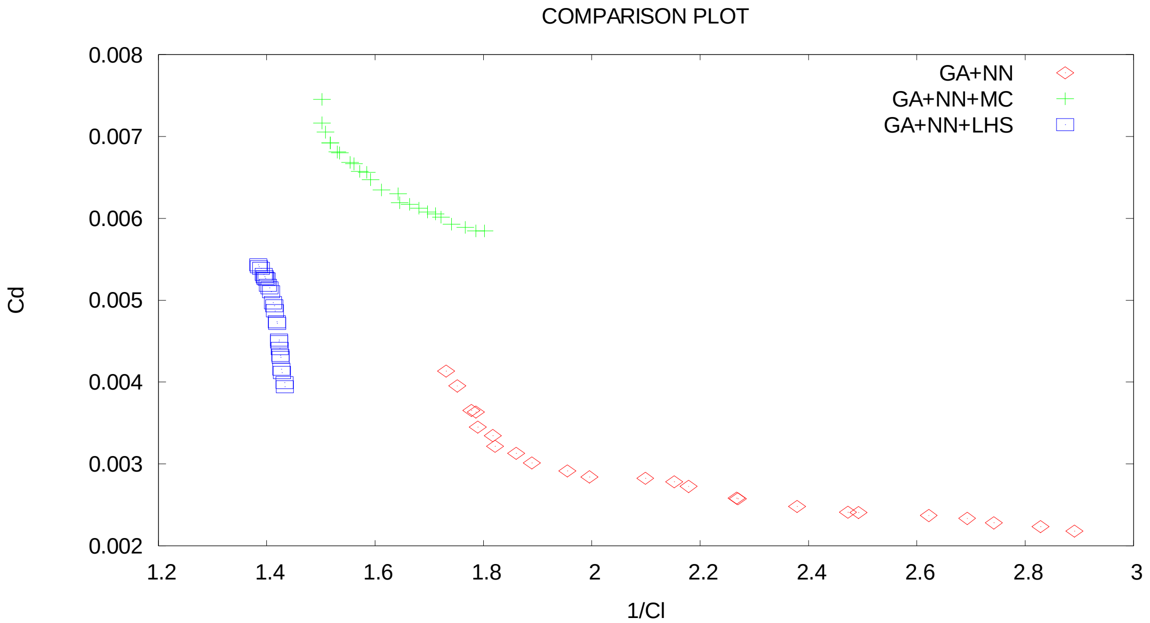

Figure 13.- Comparative plot of all three cases: Pareto fronts

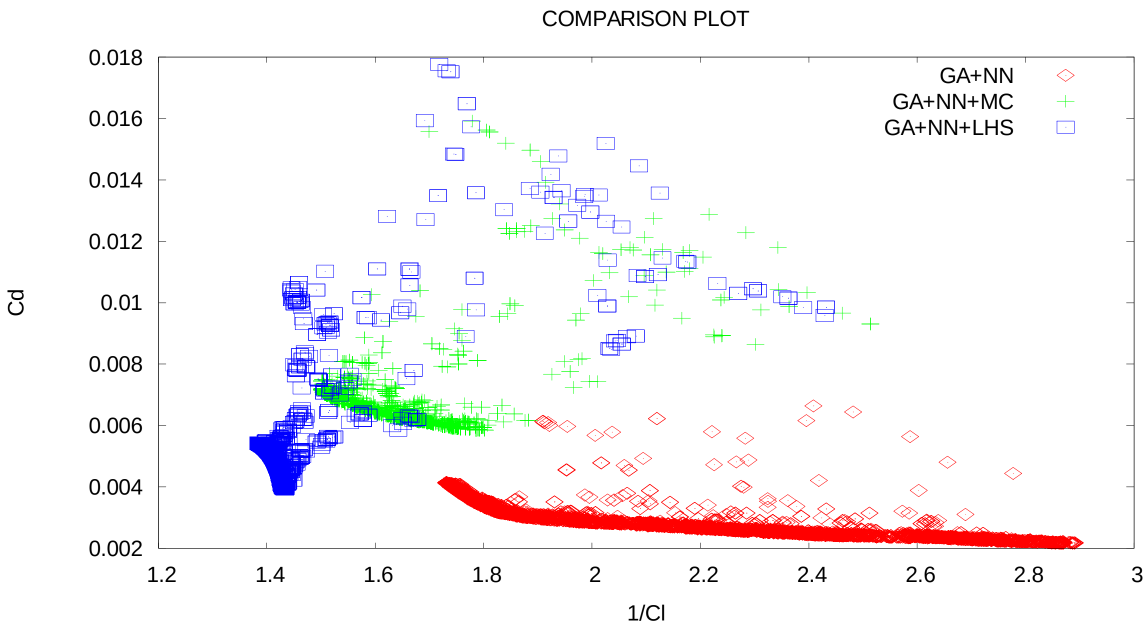

Figure 14.- Comparative plot of all three cases; whole results

The same behaviour is reproduced if we plot the best profiles; Latin Hypercube Sampling produces a set of best profiles with less dispersion. The range of best profiles obtained using genetic algorithms (GA) coupled with neural network (NN) is larger than coupling the sampling data.

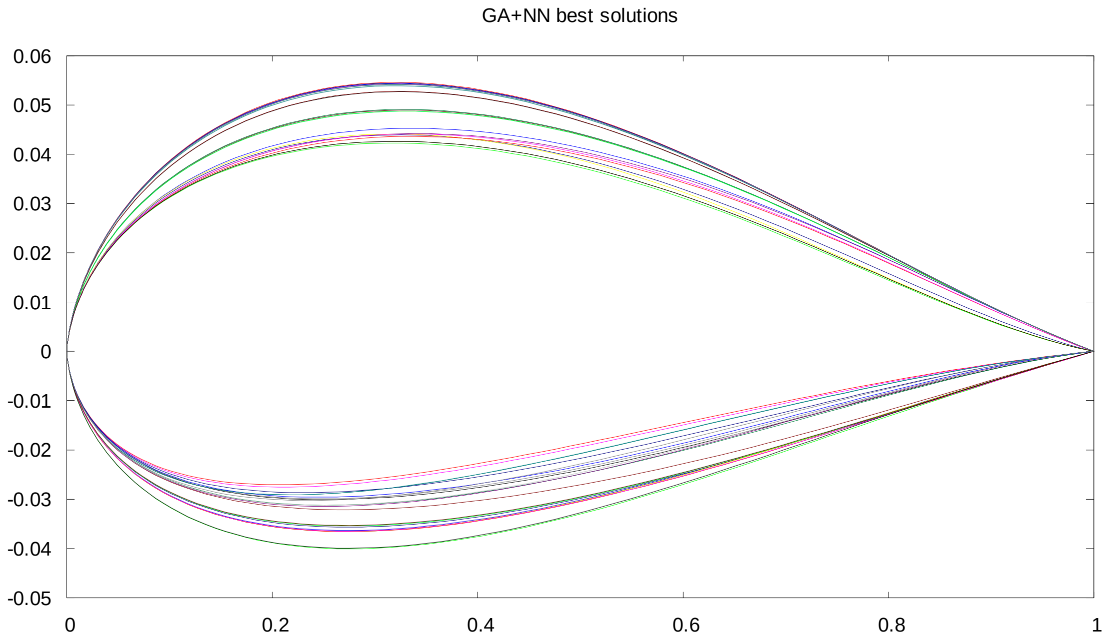

Figure 15.- Best Solutions; Genetic algorithm coupled with Neural Network

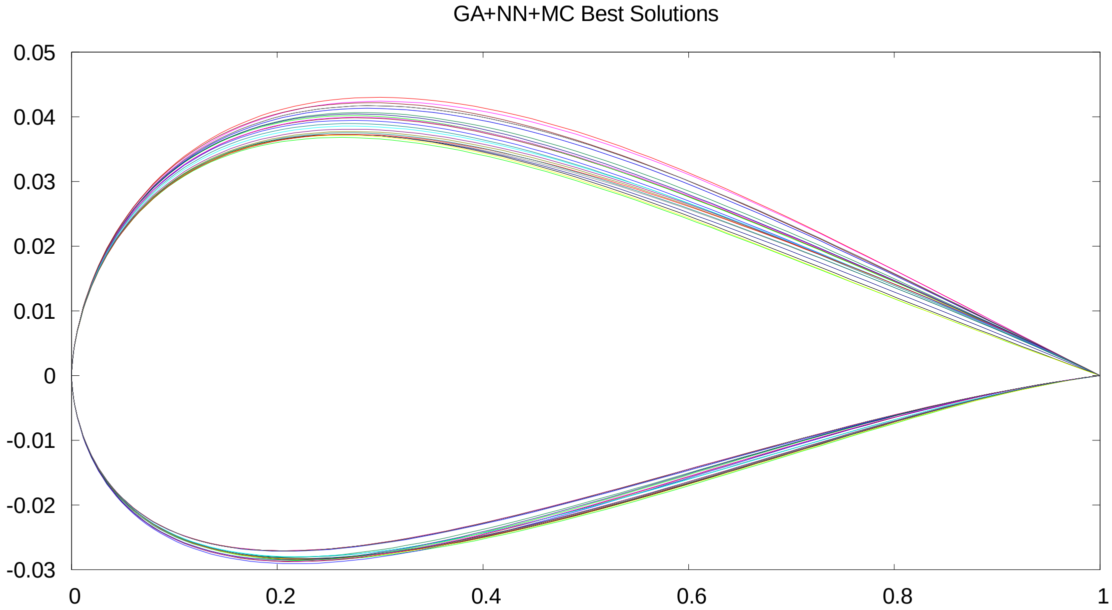

Figure 16.- Best solutions; Genetic algorithm coupled with Neural Network and MonteCarlo sampling

Comparing both sampling techniques, GA with NN and Latin Hypercube sampling technique is able to converge to the best lower profile, which is the same for all best upper profiles obtained. It does not depend on the sample values defined by LHS, because the same convergence point is obtained when using different sample data.

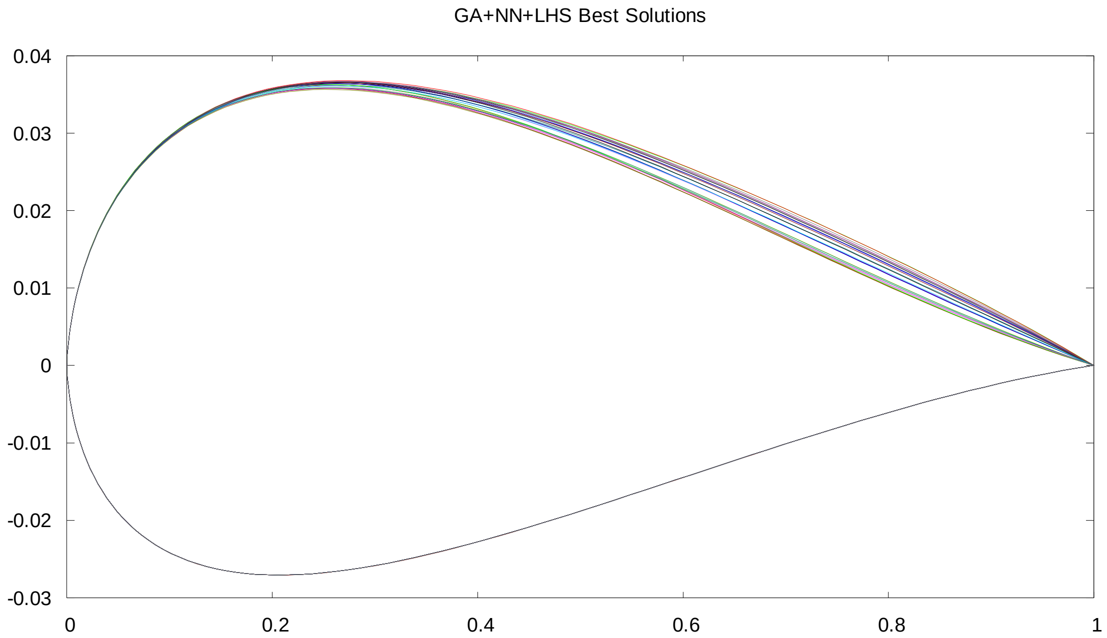

Figure 17.- Best Solutions; Genetic algorithm coupled with Neural Network and Latin Hypercube sampling

The implementation of probabilistic data into the genetic algorithm defines an improvement. In comparison with genetic algorithm probabilistic data from Latin Hypercube sampling obtains a best set of solutions. The Pareto front of this analysis is shorter, and presents a better set of solutions.

Further work should be done in other to analyse the effect of several different probabilistic data to the results. The probabilistic data can be applied in several ways, and to different values; I mean, contour conditions values, or initial values as well.

6. Acknowledgements
The authors would thank all the people in CIMNE which helped to develop this work in several ways. Special thanks to Dr. Roberto Lopez who help us in the development of the neural network, to Dr. Roberto Flores and Enrique Ortega who provided valuable help regarding PUMI solver, and also thanks to Dr Chris Lee for his appreciated comments and suggestions.
This research and all the work related is partially founded by the European Commission project CFD-NODESIM (AST5-CT-2006-030959; Contract Number 030959), and by the Spanish Ministry of Education and Science, Project DPI-2008-05250.

7. References

Books

[1] Bazaraa M.S., Sherali H.D., Shetty C.M.; *Nonlinear programming: theory and algorithms*. John Wiley & Sons, New York, 1993.

[2] Nash S.G., Sofer A., *Linear and non-linear programming*. McGraw-Hill, 1996.

[3] Davis L., *Handbook of genetic algorithms, Edited by Lawrence Davis*, Van Nostrand Reinhold, New York, 1991

[4] Lopez R., *Flood A1. An open source Neural Networks C++ Library User Guide*, CIMNE, 2007.

[5] Lopez R., Oñate E., A *variational formulation for the Multilayer Perceptron, Artificial Neural Networks*, ICANN 2006, Lecture Notes in Computer Science 4132(I) pp159-168, 2006.

Proceedings

[6] J.P. Hamalainen et al, *Evolutionary shape optimization in CFD with industrial applications*, ECOMASS, Barcelona, 2000

[7] Papadrakikis M., Lagaros N.D., Tsompanakis Y., *Structural optimization using evolution strategies and neural networks*, Comput. Methods Appl. Mech. Eng 156 (1998) pp 303-333.

[8] Plevris V., Lagaros N. D., Papadrakakis M.; *Robust design optimization of steel structures*; 5th GRACM International Congress on Computational Mechanics Limassol, 29 June – 1 July, 2005

Reports, Theses, and Individual Papers

[9] Huyse L., Michael Lewis R.; *Aerodynamic shape optimization of two-dimensional airfoils under uncertain operating conditions*, ICASE Report 2001-1, January 2001.

[10] Nagarathinam S., González L.F., Whitney E.J., Srinivas K., Périaux J.; *Aerodynamic shape optimization using a robust evolutionary algorithm and grid-free flowsolver*

[11] Marco N., Desideri J.A., Lanteri S., *Multi-objective optimization in CFD by Genetic Algorithms*, INRIA Report nº 3686, 1999.

[12] Obayashi S. el al, *Design optimization of supersonic wings using evolutionary algorihms*, ECOMASS 1998.

[13] Jeong S., Murayama M., Yamamoto K., *Efficient optimizationdesign method using Kriging Model*, AIAA 2004-118.

[14] Deb K., Pratap A., Agarwal S., Meyarivan T., *A fast and elitist multi-objective genetic algorithm: NSGA-II* , KanGAL Report 200001.

[15] Lopez R., Balsa-Canto E., Oñate E., *Neural networks for variational problems in engineering*, International Journal for Numerical Methods in Engineering, Early View, 2008.

Computer Software

[16] STAC. A software platform for stochastic analysis using Montecarlo method. CIMNE, Barcelona, 2006.

[17] Tdyn. A finite element code for multiphysics fluid dynamics problems. COMPASS (www.compassis.com), 2007.

Unpublished Papers and Books

[18] T. Back, H-P. Schwefel, *An overview of evolutionary algorithms for parameter optimization*, University of Dortmund

[19] Mitra A., *Design of transonic airfoil using genetic algorithms.*

[20] Quagliarella D., Vicini A., *Designing high-lift airfoils using Genetic Algorithms.*

[21] Obayashi S., Sasaki D., Takeguchi Y., *Evolutionary computation of supersonic wing shape optimization.*

[22] Flores R., Ortega E., *PUMI: an explicit 3D unstructured finite element solver for the Euler equations.*