Excerpt from the Proceedings of the COMSOL Conference 2010 Boston

Multi-Objective Optimization of a Strip-Fin Microchannel Heatsink: Integrating COMSOL Multiphysics with modeFRONTIER

Alberto Clarich, Rosario Russo, Nader Fateh*
Esteco srl Esteco North America
*1221 Brickell Ave, Miami, FL, 33131, USA fateh@esteco.com

Abstract: For this work the multi-objective design environment modeFRONTIER was integrated with COMSOL, thereby bringing the advantages of multi-objective optimization to multiphysics simulations. Practically any commercial analysis tool (such as COMSOL), or in-house code, can be integrated within modeFRONTIER's framework. The optimization can be either direct, employing algorithms such as Game Theory or Genetic Algorithms, or it can be virtual, where a continuous response surface (such as a neural network) is created from a set of discrete data points, and used to predict better performance in areas of design space where no data point has been calculated.
In the case of the direct approach, modeFRONTIER determines the configurations to be analyzed, and automatically launches COMSOL, monitoring the results in order to assess the performance relative to the objectives.
The multi-objective optimization of a strip-fin microchannel heat sink is shown, where the target was to simultaneously minimize both the temperature variation on the chip and the cooling pump power, while constraining the maximum temperature on the chip. Both geometric and process variables were employed; these included a mix of continuous and discrete, dimensional and non-dimensional.
modeFRONTIER's extensive post-processing toolkit, consisting of both statistical and graphical methods, was used to gain a full understanding of the results obtained, and to increase the efficiency of the optimization carried out.

Keywords: multi-objective and multi-disciplinary optimization, distributed and automatic computational environment.

1. Introduction

Multi-objective optimization, where the search process is driven by mathematical algorithms rather than by human intuition, is increasingly becoming a fundamental part of the design process. Moreover, when multiple software tools are coupled together to create an integrated, multi-disciplinary, simulation environment, the power of such methods becomes even greater.

The traditional, intuition-based, design approach (which often involves a great degree of 'trial and error') usually requires many iterations, where a designer, or a design team, will need to assess the outcome of a set of simulations from multiple aspects (e.g. FEA, CFD, acoustics, etc), before deciding how to modify the design in order to improve the overall performance. Modifications can take a significant amount of time, due to the fact that the geometry may need to be modified (e.g. in a CAD program), new meshes created, and new simulations run – all of which can take large amounts of man-time.
However, if a multi-objective, multi-disciplinary design tool such as modeFRONTIER is employed, a single, integrated, environment is created in which all the CAD/CAE tools of interest are coupled together, and the models are parameterized. This allows the user to launch an automatic optimization, where modeFRONTIER modifies the models, runs the required simulations, and extracts the outputs of interest; these in turn are used by the optimization algorithm to drive the design to try to achieve all the user-defined objectives.

2. modeFRONTIER's Multi-Objective Design Environment

This paper describes how modeFRONTIER[1] is used to integrate COMSOL Multiphysics in an optimization environment, which allows a series of simulations to be run automatically., The configurations run were determined by the selected optimization algorithm, resulting in a set of optimal solutions .

To illustrate the integration, a multi-disciplinary test case modeled in COMSOL Multiphysics was used: the optimization of a strip-fin microchannel heatsink, with the objectives of minimizing chip temperatures and pump power consumption.

In modeFRONTIER's modular environment, all components of the integration and optimization, including input variables, input files, scripts to run commercial software, output files, output variables and objectives, are defined as icons to be connected to other components.

This allows the user to define the complete logic flow, from CAD parameterization to performance evaluation, as well as to select a Design of Experiment (DOE) method, or an optimization algorithm appropriate for the number and type of objectives. The optimization algorithms available to the user include Genetic Algorithms [3], Evolutionary Algorithms, Game Strategies [4], Gradient-based Methodologies, Response Surfaces and Robust Design Optimization. The user also has an extensive range of DOE algorithms to choose from: Sobol, Factorials, Latin Square, Montecarlo, D-Optimal, etc.

3. The Strip-Fin Micro-Channel Heatsink Problem

On-going improvements in micro-electronic device performance, with the requirement of ever higher circuit densities and operating speeds, are leading to increasingly greater challenges related to component cooling. It is anticipated that future generations of microprocessors and microelectronic components will have to dissipate heat fluxes in excess of 1000 W/cm^2, which will necessitate the introduction of new types of cooling systems. One option may be to turn to liquid cooled systems, with a coolant flowing across rows of microchannel heatsinks. While this approach will undoubtedly lead to more efficient cooling in terms of overall heat transfer, high temperature gradients along the chip could lead to unacceptable variation of chip performance with local temperature, as well as the introduction of undesirable internal stresses.

In order to investigate this problem, a test model of a micro-channel heat sink [2] was set up, consisting of a matrix of micro strip fins, which are repeated over a surface of 12 mm by 12 mm, and with a total height of 0.5 mm, as shown in Figure 1. A uniform heat source of 100 W/ m^2 was applied to the bottom surface and an inlet and outlet for the coolant were applied at the front and back of the domain. In order to take advantage of the cyclic nature of this problem, only a single channel was modeled, as shown in Figure 2, with symmetry boundary conditions applied on either side. The flow was solved in COMSOL by coupling weakly compressible Navier-Stokes equations for the water flow with general heat transfer models. Figure 2 also shows an example of a velocity and a temperature field.

The objective of the optimization was to find a microchannel shape which would:

- Minimize the temperature delta (difference between the maximum and minimum values) over the heatsink base, corresponding to the chip surface temperature, for better cpu performance and lifetime [2]
- Constrain the maximum temperature of the heatsink, to be lower than 344 K
- Minimize the coolant pump power, to reduce energy consumption

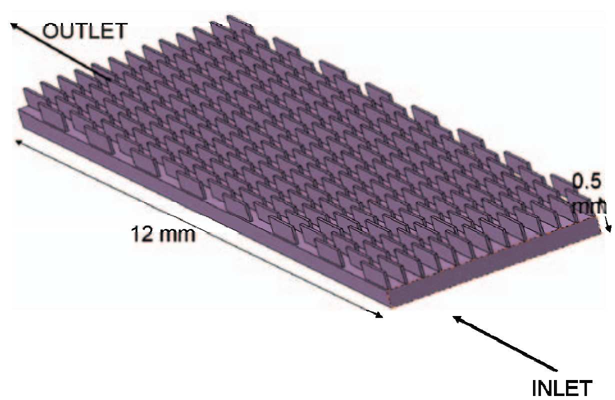

Figure 1. COMSOL Model with Boundary Definitions

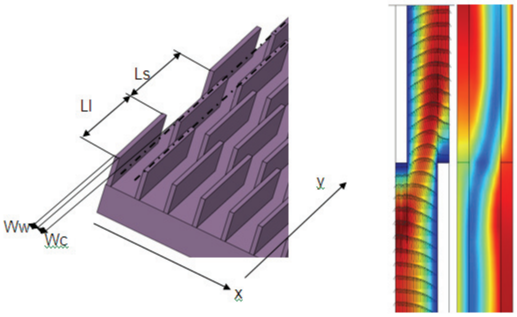

Figure 2. COMSOL Model showing Parameter Definition and Velocity/Temperature Fields

5 geometrical parameters were defined directly in the COMSOL model to modify the fin shape, the number of fins, and the water mass flow:

- Kx = Ww/Wc (ratio of fin width to channel width)
- Ky = Ll/Ls (ratio of fin length to inter-fin gap length)
- Nx = number of fins along x
- Ny = number of fins along y
- M = mass flow [kg/s]

4. Integration of the COMSOL Model in modeFRONTIER

In order to integrate the COMSOL model in modeFRONTIER, a parametric file defining the model [8] needs to be created. This is done automatically by saving the COMSOL model in the .m format.

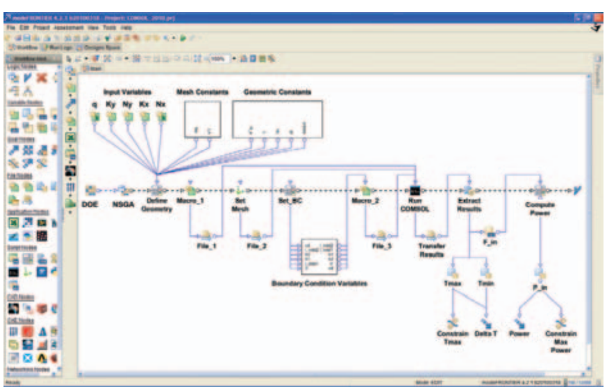

Figure 3. The modeFRONTIER Optimization Workflow

modeFRONTIER can now be used to create a workflow, in which all the processes necessary to run the case, and hence the optimization, are linked together. Figure 3 shows this workflow; the nodes representing the input variables can be seen at the top left, while the 2 boxes to the right of those contain constant parameters (which will not vary during the optimization). The input variables were defined with lower and upper limits, as well as with discretization step sizes.

When COMSOL runs in batch, it reads an input file which contains all the variables. Once integrated with modeFRONTIER, this input file will be updated by modeFRONTIER with each new configuration which arises during the optimization. Therefore, modeFRONTIER sets the values of the variables (either according to the DOE scheme being used, or to the optimization algorithm search direction), updates the COMSOL input file, and then runs COMSOL in batch to read that input file.

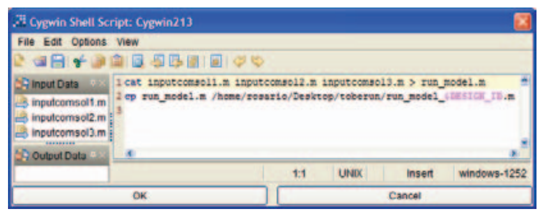

Figure 4. COMSOL Batch Command File

Figure 4 shows the contents of the "Run_COMSOL" node in Figure 3, where COMSOL is run in batch. The last command of the *model.m* file is

'*save output.dat -ASCII –DOUBLE*'

which forces COMSOL to write all the output variables of the model in an ASCII file, in this case called *output.dat.*

The output variables of interest in this case are the maximum and minimum temperatures on the heatsink base surface, extracted directly from the file as scalar values, and used to define the delta temperature objective and the maximum temperature constraint, as well as the integral of the pressure over the inlet and the outlet, from which he overall power consumption can be calculated.

5. Optimization Strategy in modeFRONTIER

In order to reduce the overall number of designs needed to optimize the heatsink, an efficient optimization strategy was adopted in modeFRONTIER, consisting of the following steps:

1) A preliminary exploration using a Design of Experiments (DOE) to sample different configurations, with the goal of learning as much as possible about the system from as few designs as possible

2) Using modeFRONTIER's statistical analysis tools to understand which part of design space gives the most promising results

3) Use of the most appropriate optimization algorithm in modeFRONTIER to find the optimal solutions.

The first step therefore is the definition of a suitable DOE; several algorithms are available in modeFRONTIER, including random, quasi-random (SOBOL), factorial algorithms, Taguchi, Montecarlo-Latin Hypercube, incremental space filler and D-Optimal. In this case, in order to have a wide and uniform exploration of design space, a Uniform Latin Hypercube (ULH) DOE [6] of 50 designs was chosen, guaranteeing a uniform distribution for all input variables, each with a high number of levels.

Figure 5 below shows a modeFRONTIER Scatter Matrix created for the input variables' parametric values to be run in the DOE. The correlations for each pair of input variables are shown below the diagonal, with values ranging from -1 to +1; values close to +1 show that the variables are highly directly correlated, -1 indicates that the variables are highly inversely correlated, while values close to zero suggest low correlation between that pair of variables. As the parameters were uniformly distributed in design space we expect all the correlation to be low, as confirmed by Figure 5. The scatter charts above the diagonal in Figure 5 further confirm this: the parameter values are all well dispersed with no clustering (which would have led to higher correlations). The distribution plots along the diagonal also show that there is a uniform spread of values of the parameters across their ranges.

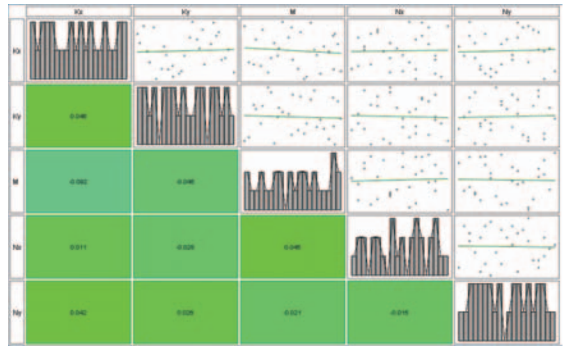

Figure 5. modeFRONTIER Scatter Chart of Parameter Values

Figure 5 confirms that the ULH method chosen will give statistically unbiased data, allowing us to use the statistical analysis tools in modeFRONTIER on the results with confidence. At this point all configurations specified by the DOE (ULH) were run automatically through the modeFRONTIER workflow. Once these designs were executed, a correlation matrix was created to show the relationships between the input variables and the outputs, as shown in Figure 6.

From this matrix we can conclude:

- The variable Nx (the number of fins in the transverse direction) seems to be insignificant for all the objectives, since the correlations are very low: this allows us to set Nx to a constant value in the optimization.

- The variable Ny (the number of fins in the flow direction) can be seen to have a fairly high direct correlation with the power objective; therefore a high value of Ny leads to a high value of power. Since the objective was to minimize power, it was decided to restrict the value of Ny to the lower half of its range. Since Ny has a low correlation with the other objective (delta_T), this should not adversely affect the achievement of that goal.

- The variable M (mass flow) can be seen to have more effect on delta_T than on power; moreover, high values of M lead to lower values of delta_T (since there is an inverse correlation).

- The variables Kx and Ky (ratio of fin dimensions) both have opposite effects on the objectives: high values reduce delta_T (desirable), but increase power (undesirable); therefore it was decided to keep the full range of variation for these two parameters.

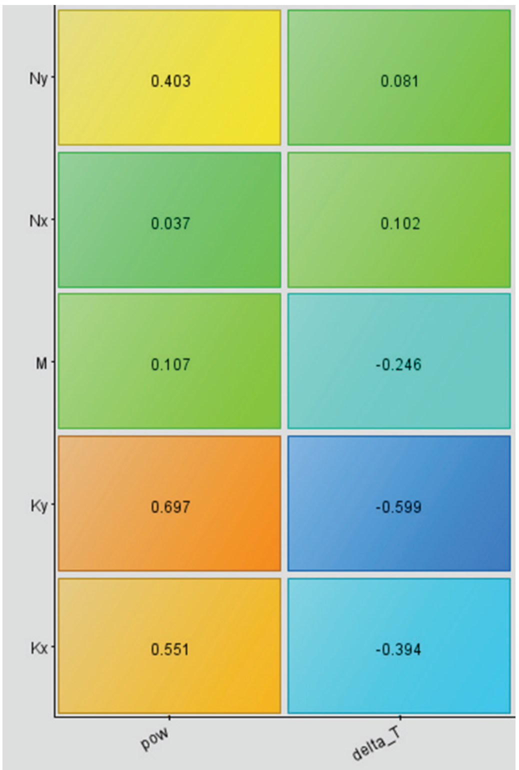

Figure 6. Correlation Matrix of Inputs vs Objectives

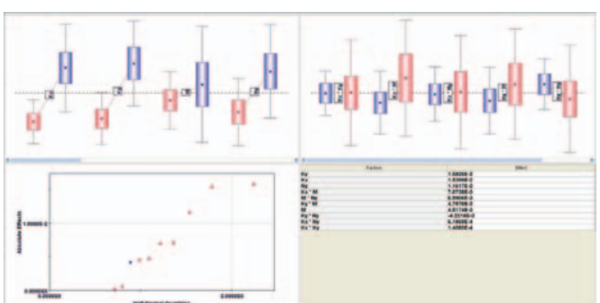

Figure 7. Student Chart showing Interaction Effects

A more accurate analysis can be obtained using Student charts [11], as shown in Figure 7 for the power objective. The effect size for each variable is shown in the upper-left box of the chart, which also includes statistical information on the population variance (lower ranges of the variable are in red, higher ranges in blue), while the upper-right box also shows the effect of the interactions between the variables (only mass seems to have important interactions with the other factors). The two lower boxes show the half-normal quantiles and the effects summary table. The conclusions are similar to those reached from study of the correlation matrix.

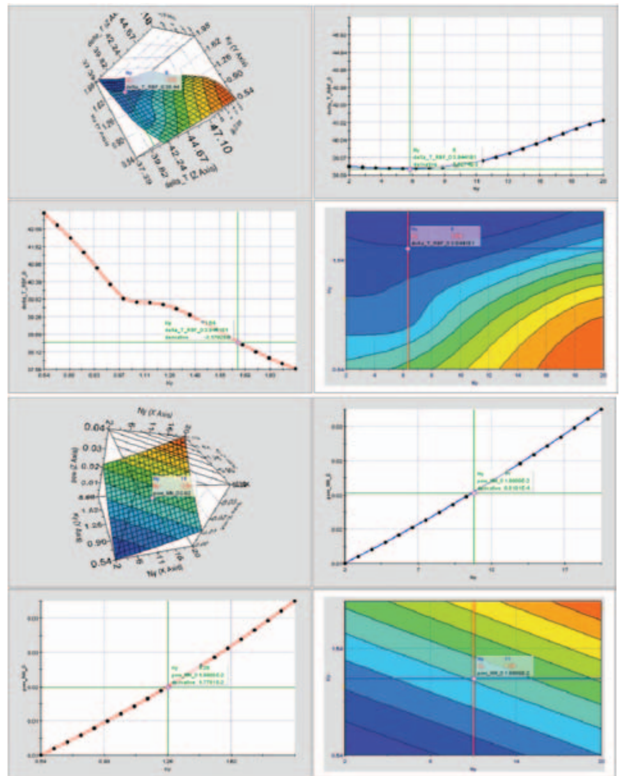

Figure 8. Response Surfaces for Delta_T (top) and Power (bottom)

Another powerful tool in modeFRONTIER to understand the behavior of the system is RSM (Response Surface Methodology) [12]. The computed designs can be used to train a mathematical metamodel, in order to have an interpolation of the outputs as a function of the inputs. Therefore, we start with a set of discrete data points which are used to create a continuous response surface, which in turn can be used for optimization. Several RSM tools are available in modeFRONTIER, including Kriging, Radial Basis Functions, Neural Networks, Genetic Programming, Polynomials, Gaussian Functions, etc. Once the RSM has been trained, the effect of the inputs on the outputs can be visualized. Figure 8 shows the two most influent variables, Ny and Ky plotted against each of the two objectives, confirming the strong correlations and the interactions.

Another tool, Clustering [9], can be used to identify the optimal ranges for the input variables. Clustering consists of finding group of designs with similar properties in terms of inputs and outputs, and several algorithms are available in modeFRONTIER, ranging from hierarchical clustering to K-Mean clustering. Applying the former, with the Ward approach, it was possible to identify two clusters, one with the best values of power and the other favoring delta_T. These two clusters can be seen in the parallel chart in Figure 9: each input and output variable is represented by a vertical axis, and the two clusters are represented by two bands: the central line gives the value of each variable for the centroid of the cluster, i.e. the average value for all the points in the cluster, and the width of the band gives the range of variable values for each cluster. It can be seen that all 4 input variables have contrasting behavior, since the cluster with higher values of these variables leads to low values of delta_T, but at the expense of high power, while the other cluster (lower values of the variables) strongly favors low power values, but leads to the highest values of the the other objective (delta_T).

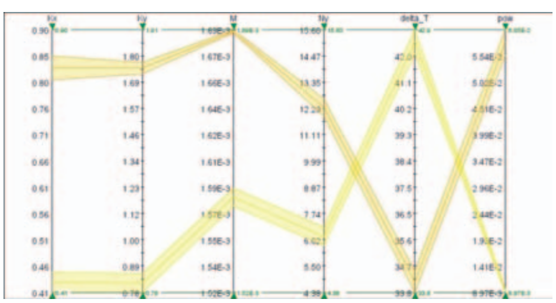

Figure 9. Cluster Distributions

It was therefore decided to reduce the range of variation of the variables, as summarized in the table in Figure 10 below.

	Nx	Ny	Kx	Ky	M
Original bounds	[110-125]	[2-20]	[0.25-1]	[0.5-2]	[0.014-0.016]
Bounds after analysis	120 (constant)	[2-16]	[0.25-1]	[0.5-2]	[0.015-0.017]

Figure 10. Range of Variation Reduction

In this reduced variable space, an optimization algorithm (a genetic algorithm) was applied with high efficiency, i.e. finding the best results with the lowest number of designs, as described below.

6 Optimization Results in modeFRONTIER

The last step of the optimization strategy is therefore to specify the algorithm (in this case, the genetic algorithm, NSGAII [10]), running 10 generations and with a population size of 10 designs (leading to a total of 100 simulations). These numbers are quite low for a typical GA, but are justified by the statistical analyses described above, which allowed an increase in efficiency by restricting the variable ranges to the most promising parts of design space. This process in effect led to a 'jump' in the evolution process, allowing optimal designs to be reached with a relatively low number of simulations.

Figure 11 below shows the result of the optimization: with delta_T plotted on the x axis, and power on the y axis, and each point representing a different configuration of the heatsink: the yellow points are unfeasible, i.e. they break at least one of the constraints (on maximum temperature and maximum power), while the blue are feasible.

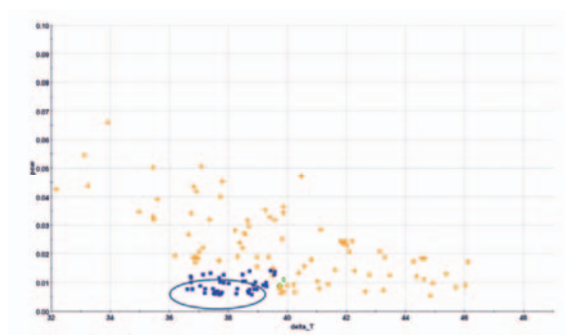

Figure 11. Optimization Results

The original configuration is indicated by a number 0 (in green): it can be seen that several points, circled, meet all constraints and dominate design 0, i.e. they give better values for both objectives.

Considering a trade-off of the two objectives, one design, was elected, as shown in Figure 12, where it is compared with the original. The Figure shows a section of the flowfield, as well as the overall temperature distribution for each case.

The table in Figure 13 gives the values of the input parameters and objectives for each of the two cases: as can be seen, the main difference between the two configurations is the number of fins in the y direction (Ny) which was reduced significantly.

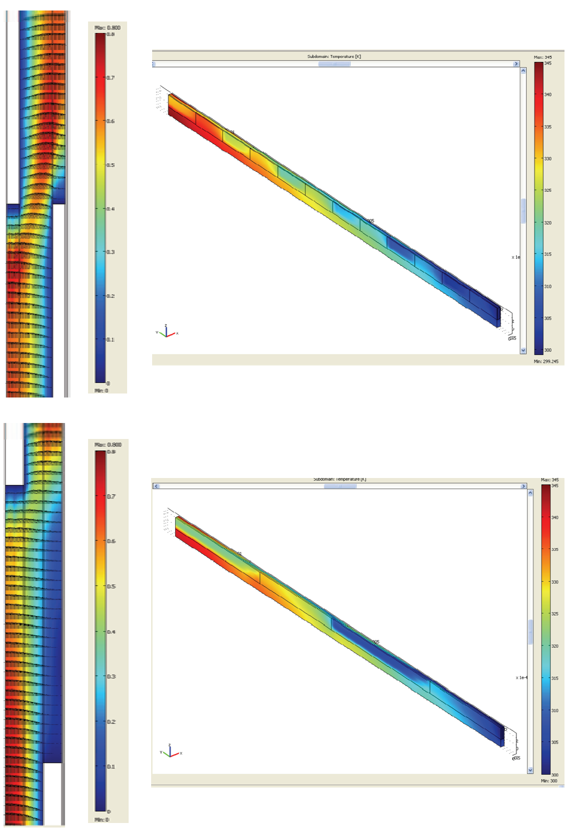

Figure 12. Original Configuration (top) and Optimized

	Nx	Ny	Kx	Ky	M	Delta T	power
Original configuration	120	14	0.969	0.726	1.57E-3	36.7	1.22E-2
Optimized configuration	120	4	0.95	0.78	1.58E-3	36.2	7.89E-3

Figure 13. Original and Optimized Configurations

6. Conclusions

This paper has shown how it is possible to integrate COMSOL in the multi-objective optimization environment modeFRONTIER. Using as an example a strip-fin microchannel heatsink optimization problem, it was shown that the efficiency of running multi-objective optimizations can be greatly increased, even with genetic algorithms, by careful selection of the initial set of configurations (DOE), and use of advanced statistical analysis tools to restrict the problem to an area of design space most likely to produce good results.

8. References

1. http://www.esteco.com
2. Fangjun Hong, Ping Cheng, *Three dimensional numerical analyses and optimization of offset strip-fin microchannel heat sinks*, International Communications in Heat and Mass Transfer 36 (2009) 651–656
3. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans Evol Computational Journal*, **Vol.6** (2002)
4. A. Clarich, V. Pediroda, C. Poloni, A competitive Game Approach for Multi-objective Robust Design Optimization, *AIAA 2004-6511, Chicago*, (2004)
5. O.L. Davies, The design and analysis of industrial experiments, Oliver & Boyd, (1963)
6 McKay M.D., Conover W. J. and Beckman, R. J., “A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code”, from Latin Hypercube Sampling (1979), Technometrics 21(2), pp. 239-245
7 Kaufman, L. and Rousseeuw, P. J. Finding Groups in Data: An Introduction to Cluster Analysis. New York, Wiley, 1990
8 A. Clarich, C. Poloni, *Multi Objective Optimisation Integrating COMSOL*

Multiphysics with modeFRONTIER, Grenoble, October 2007

9 Clarich A., Geremia P., Parashar S., Russo R., *Use of Multi-Variate-Data-Analysis Techniques in modeFRONTIER for Efficient Optimization and Decision Making,* AIAA, September 2010

10 Clarich A., Di Stefano D., Fuligno L., Otte T., *Hybrid NSGA-SOM multi-objective optimization of a medical device flowfield,* European Journal of Industrial Engineering, June 2009

11 Alberto Clarich, Valentino Pediroda, Giovanni Mosetti, Carlo Poloni**,** *Application of evolutive algorithms and statistical analysis in the numerical optimisation of an axial compressor*, The 9th of International Symposium on Transport Phenomena and Dynamics of Rotating Machinery,Honolulu, Hawaii, February 10-14, 2002

12 A.Giassi, V.Pediroda, C.Poloni, A.Clarich, *Three-dimensional inverse design of turbomachinery stage with neural-networks and direct flow solver*, International Journal on Inverse Problem in Engineering, Vol.11, number 6, Taylor & Francis, pg. 457-470, December 2003