

Multi-objective shape optimization of a heat exchanger using parallel genetic algorithms

Renan Hilbert¹, Gábor Janiga^{*}, Romain Baron², Dominique Thévenin

*Laboratory of Fluid Dynamics and Technical Flows – LSS/ISUT,
University of Magdeburg “Otto von Guericke”, Universitätsplatz 2, D-39106 Magdeburg, Germany*

Received 13 July 2005; received in revised form 14 November 2005
Available online 13 March 2006

Abstract

We perform in this paper a multi-objective design optimization concerning the blade shape of a heat exchanger, considering the coupled solution of the flow/heat transfer processes. For this, a genetic algorithm is used. The aim of the procedure is to find the geometry most favorable to simultaneously maximize heat exchange while obtaining a minimum pressure loss. An in-house computer package, called OPAL, performs the optimization process in a fully automatic manner. It calls the pre-processor to generate the computational geometry as well as the mesh, it then performs the numerical simulation of the coupled fluid flow/heat transfer problem using Fluent, calculates the output parameters, and iterates the procedure. The genetic algorithm relies on a relatively large number of simulations, which may result in a considerable computational effort, depending on the configuration. The procedure can thus be performed in parallel on a Linux PC cluster to reduce user waiting time. A nearly optimal speed-up is obtained for the present configuration.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Heat exchangers; Laminar; Optimization

1. Introduction

Designing optimal shapes for practical engineering applications has been the subject of numerous publications during the last decade [1–4]. Many methods can be found in the literature for optimization problems, based on different strategies, most of the time developed for a specific class of models. In this project, we consider specifically *multi-objective* optimization problems, since it covers many interesting application fields. As a matter of fact, most of the time, engineers responsible for the design of industrial devices have to face problems with more than one objective to fulfill at the same time. Moreover, the objectives of the optimization process are often concurrent (a simple example being the quality/price trade-off).

Thus, the aim of the present study is to develop and test numerical tools that can be used to solve multi-objective problems, involving complex flow simulations and using computational fluid dynamics (CFD) codes for practical industrial configurations, while keeping reasonable overall computing times.

Optimization involving CFD computations is an intensive field of research. In aerodynamics an optimal shape geometry is often needed. Thus, the design of an airfoil shape was for example optimized in [1,2]. The wall shapes of incompressible diffusers have been investigated by CFD and optimized in [5]. Han and Maeng [3] have presented a shape optimization of cut-off in a multi-blade fan/scroll system analyzed using two-dimensional CFD.

In the same way, heat transfer problems often involve optimization. Heat exchange through smooth and corrugated walls has been investigated in [6]. Shape design of a cylinder with heat transfer was carried out in [7]. The optimal shapes of fins and pins inside heat exchangers are examined by various authors [4,8–11]. Tiwari et al. [12]

^{*} Corresponding author. Tel.: +49 391 67 18196; fax: +49 391 67 12840.
E-mail address: Gabor.Janiga@vst.uni-magdeburg.de (G. Janiga).

¹ Present address: ReNuDa, Paris, France.

² Present address: Diva-NG, Sophia-Antipolis, France.

Nomenclature

H	inter-blade channel width (mm)
N	number of individuals
P	pressure (Pa)
T	temperature (K)
v	velocity (m/s)
x	spatial coordinate (m)
y	spatial coordinate (m)

Greek symbol

Δ	difference
----------	------------

Subscripts

inlet	inlet
obj	objectives
param	parameters
proc	processors

have studied different angles of attack for the delta winglets mounted on the fin-surface on top of oval-shaped tubes. Heat transfer of finned and non-finned circular and elliptic tubular arrangements are investigated numerically in [13] to maximize the total heat transfer rate. The flow through a heated pipe with an inserted twisted tape was examined in [14] for different slopes. This analysis is based on the entropy production minimization [15]. Multi-parameter optimization coupled with CFD was investigated in [16] to maximize the performance of a heat sink. Okabe et al. [17] have obtained optimal results for a micro-heat-exchanger based on different multi-objective optimization methods. As a whole, optimization of configurations involving the coupled simulation of flow and heat transfer remains a fairly new field of research.

Classical optimization techniques, like gradient-based methods are known for their lack of robustness and for their tendency to fall into local optima. Generic and robust search methods, such as Genetic Algorithms (GA) [18,19], offer several attractive features and have been used widely for design shape optimization [20–24]. They can in particular be used for multi-objective multi-parameter problems. They have been successfully tested in many practical cases, for example for design shape optimization in aerodynamics [1,2,21–23], automotive industry [25]. The basic idea associated with the GA approach is to search for optimal solutions using an analogy to evolutionary theory. During the iteration (or “evolution” using GA terminology) procedure, the decision variables or genes are manipulated using various operators (crossover, mutation, ...) to create new design populations, i.e., new sets of decision variables. The use of a fully automatic Genetic Algorithm coupled with CFD for a multi-objective heat transfer problem still remains limited by the computing time, and is up to now far from being a practical tool for engineering applications.

The purpose of this paper is to illustrate a possible methodology for the fully automatic optimization of a heat exchanger. We are neither interested here in developing a new algorithm for optimization nor to discover a completely new heat exchanger structure. We solely wish to demonstrate that it is possible to reach an optimal geometrical configuration for a case involving coupled fluid flow

and heat transfer phenomena, investigated using CFD with a reasonable computing time, for configurations very close to practical ones. In a first step, we consider laminar flows, because it corresponds to a realistic engineering problem, for example for low-power systems. Therefore, we choose a model problem, consisting of a tube bank heat exchanger. The problem is to optimize the shape of the blades so that the heat exchange is maximal while keeping a minimal pressure loss. A set of automatized numerical tools are used together to solve this problem, involving mesh generation, CFD, an in-house C++ implementation of Genetic Algorithms, a shell-script and complementary C programs for automatization and parallelization.

In what follows, the model problem is introduced first, putting into evidence the requirements for the choice of an adequate optimization strategy. The used multi-objective genetic algorithm, based on the concept of Pareto dominance, is then described. The practical computational methodology for mesh generation, CFD solution and parallelization is presented afterwards. Results are then shown and discussed, followed by concluding remarks.

2. Problem description

2.1. Tube bank heat exchanger

As an example of what can be done by coupling parallel Genetic Algorithms and CFD codes, a two-dimensional model of a tube bank heat exchanger is considered here. The simulated staggered configuration is shown in Fig. 1. Air enters the domain at $T_{\text{inlet}} = 293$ K and is warmed up by passing through the blades in which a warm fluid flows in the corresponding practical application. The blades are supposed to have a constant outer wall temperature, $T_{\text{wall}} = 353$ K. The outlet is at atmospheric pressure.

The optimization problem consists of finding the best geometry of the blades to increase heat exchange while at the same time to limit the pressure loss. The two corresponding numerical parameters to optimize are the average temperature difference ΔT and pressure difference ΔP . These two objectives are obviously inter-related. If the exchange surface increases, the heat exchange will be

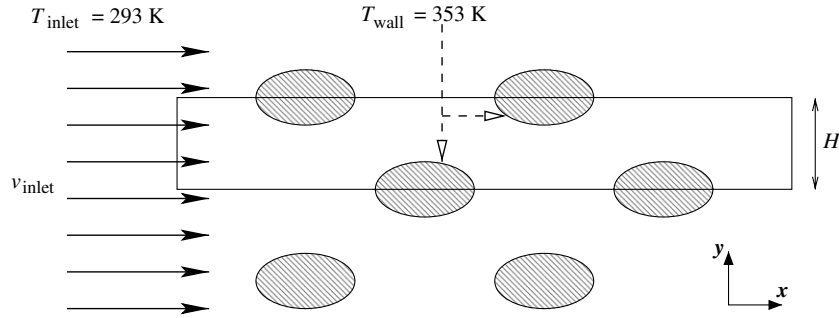


Fig. 1. Schematic description of the tube bank heat exchanger configuration considered in this work.

favored and the temperature difference between inflow and outflow will be also enhanced. But, simultaneously, for a given air flow rate the pressure loss will increase, and the heat exchanger loses efficiency. In this first examination the hydraulic resistance inside the tubes and the pumping power are not considered, but it would be an interesting subject for future investigations.

The domain bounded by a black line in Fig. 1 is simulated in this study. The Reynolds number is equal to 170 defined using the inter-blade channel width $H = 5$ mm and the uniform velocity at the inlet, $v_{\text{inlet}} = 0.5$ m/s. The length of the domain has been chosen to prevent any influence of the inflow or outflow boundary conditions on the inter-blade flow. Corresponding tests have in particular demonstrated the importance of extending the computational domain well beyond the last blade in order to avoid the influence of the boundary conditions. The full extent of the numerical domain can be seen in Fig. 2.

In this configuration, the flow is laminar, which corresponds to practical low-power applications. Computing the flow as a steady two-dimensional flow is in this case a very acceptable approximation of the true physics, as the blade length in the z direction is very large compared to its width.

2.2. Problem parameters

For the different simulations, the boundary and inlet conditions are the same, only the computational geometries differ. The outer dimensions of the computational domain as well as the blade positions along the domain boundaries are fixed and only the shape of the blades inside the computational region is varied. The forms of all four blades are always changed simultaneously, so that they are identical in every individual computation. Their geometrical shape is prescribed using four parameters, as presented in Section 4.2.1. Since all other properties and boundary con-

ditions are constant, these four parameters are the only input parameters of the optimization algorithm. After defining the computational geometry and obtaining a corresponding mesh, the numerical simulation can be performed. In this study, we use the commercial computational fluid dynamics program Fluent 6.1 [26] to solve the governing equations of the fluid flow phenomena including the energy equation. The two-dimensional fields of pressure and temperature are obtained in this way, and provide the two objective parameters, the temperature and pressure differences: ΔT , ΔP .

Genetic algorithms require a large number of simulations, leading to a high computational effort, because they operate on the entire allowed design space. Due to the fact that the objective values associated with each set of design parameters can be evaluated independently, a possibility to speed-up the optimization for the user is to perform the numerical simulations in parallel, using several processors, as described later. In the following section, the optimization method is explained first.

3. Genetic algorithms for multi-objective optimization

3.1. Multi-objective optimization

Mathematically speaking, a multi-objective problem consists of optimizing (i.e. minimizing or maximizing) several objectives simultaneously, with a number of inequality or equality constraints. The problem can be formally written as follows:

Find $\mathbf{x} = (x_i) \forall i = 1, 2, \dots, N_{\text{param}}$ such as

$f_i(\mathbf{x})$ is a minimum (respectively maximum) $\forall i = 1, 2, \dots, N_{\text{obj}}$

subject to:

$$g_j(\mathbf{x}) = 0 \quad \forall j = 1, 2, \dots, M, \quad (1)$$

$$h_k(\mathbf{x}) \leq 0 \quad \forall k = 1, 2, \dots, K, \quad (2)$$

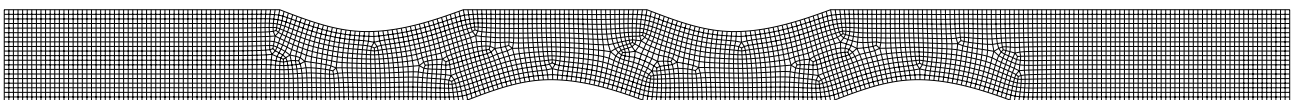


Fig. 2. Typical computational grid.

where \mathbf{x} is a vector containing the N_{param} design parameters, $(f_i)_{i=1,\dots,N_{\text{obj}}}$ the objective functions and N_{obj} the number of objectives. In this study, only inequality constraints are considered and are prescribed as bounded domains. In other words, upper and lower limits are imposed on all parameters:

$$x_i \in [x_{i,\text{min}}; x_{i,\text{max}}], \quad i = 1, \dots, N_{\text{obj}}. \quad (3)$$

The objective function $(f_i(\mathbf{x}))_{i=1,\dots,N_{\text{obj}}}$ returns a vector containing the set of N_{obj} values associated with the elementary objectives to be optimized simultaneously. In the current case, four design parameters ($N_{\text{param}} = 4$) and two objectives ($N_{\text{obj}} = 2$) are considered.

A common practice to solve such a problem is to use a trade-off between the objectives by linearly combining them using some fixed weights prescribed by the user (see for example [19,27]). The resulting single objective function is optimized using either a classical gradient-based or simplex method [28]. The first limitation of this kind of approach is that the choice of the weights associated with each objective obviously changes the solution of the optimization problem. A bad choice can lead to completely sub-optimal results in comparison with the solution obtained by considering the inter-related objectives in an independent manner. Moreover, this method does not allow to access all the set of optimal solutions (see Section 3.2).

The GAs are semi-stochastic methods, based on an analogy with Darwin’s laws of natural selection [18]. Each configuration \mathbf{x} is considered as an individual. The parameters x_i represent its genes. The main principle is to consider a so-called population of N individuals, i.e. a set of individuals covering the search domain, and to let it evolve along generations (or iterations) so that the best individuals survive and have offsprings, i.e. are taken into account and allow to find better and better configurations.

The characteristics of the GA used in the present study are based on the approach proposed by Fonseca and Fleming [29]. The genes (sometimes called characters) of the individuals (also called strings or chromosomes) are the N_{param} design parameters, encoded using a floating-point representation [30]. The initial population is a set of randomly chosen configurations in the domain defined by the limits imposed on the parameters, Eq. (3). The creation of a new generation from the previous one is performed by applying genetic operators to the individuals of the present generation, as described below. At each generation the individuals are classified as a function of their corresponding objective values, leading to a rank within the population and finally to a fitness. The definition of the rank for our specific case is described later in Section 3.2. The probability for an individual to participate in the reproduction process is determined by a probability based on its fitness value, linearly calculated from its rank in the classification. For example, for individual number i in a group of N individuals:

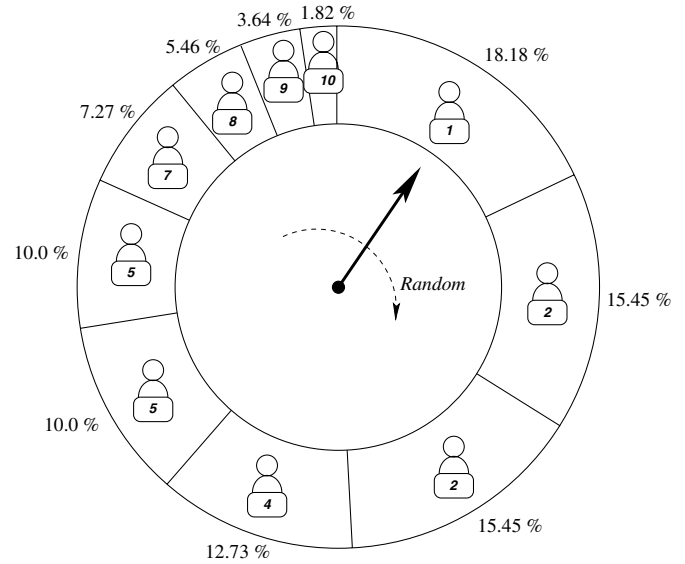


Fig. 3. Example for the rank of 10 individuals and the corresponding probability (Eq. (4)) to participate in the reproduction process, represented on a circular diagram.

$$\text{Fitness}(i) = \frac{N - \text{rank}(i) + 1}{\sum_j (N - \text{rank}(j) + 1)}, \quad (4)$$

with index j varying from 1 to N .

Fig. 3 depicts a simple example showing for a group of 10 individuals the rank values and the corresponding probability to participate in the reproduction process, directly based on its fitness. Using this technique individuals with equal rank have an equal probability to reproduce. Individuals associated with a higher fitness value have a better chance to survive and to take part in the reproduction process, as explained in Section 3.3. In this way better and better generations are generated step by step. GAs operate on the entire population. Thus, they offer a good potential to explore the whole search space and to avoid local optima. Their good robustness is mainly due to the fact that there is no evaluation of the objective function’s derivatives. Moreover, the process can iterate further even if some evaluations fail. The main drawback associated to evolutionary algorithms in general remains their cost in terms of computing (CPU) time. But, due to the fact that the evaluations are performed independently, they are easily parallelizable, as shown later.

3.2. The concept of Pareto dominance

In a multi-objective problem, the set of parameters (the individuals in the GA terminology) can be compared according to Pareto’s rule [29]: the individual A dominates the individual B if, for at least one of the objectives, A is strictly better adapted than B and if, for all other objectives, A is not worse than B. An individual will be considered as optimal if it is non-dominated in the sense of this rule. The rank is computed for each individual according

to the number of individuals dominating him. If an individual is not dominated by any other individual, he gets the top rank (of course, there may be several non-dominated individuals at the same time). Then come all individuals that are dominated by only one individual, and so on. This is a true multi-objective approach because objectives are considered as independent and there is no trade-off. An individual i that is dominated by j individuals is given $\text{rank}(i) = 1 + j$. Non-dominated individuals have rank 1, so $1 \leq \text{rank}(i) \leq N$. At the end, the best individuals are all the non-dominated individuals over all generations, leading to the “computed” Pareto front, supposed to represent the real one. The use of this rule allows one to classify the individuals of the population and therefore to calculate the corresponding fitness values, Eq. (4).

3.3. A genetic algorithm for multi-objective problems

Three groups are defined in the population, two for the Genetic Algorithm generations and one for storing the non-dominated configurations:

Elite: The currently non-dominated individuals.

Parents: Individuals that may reproduce.

Offspring: Individuals of next generation, built from parents.

An individual can belong at the same time to several groups. To generate an offspring, one or two parents are selected using their fitness values. The selection of the parents relies on the roulette-wheel method. Each roulette-wheel slot receives a current individual from the population. An individual with better fitness value is associated with a larger roulette-wheel slot size (Fig. 3). A larger size of a roulette-wheel slot for a maximization problem correspond to a better chance to survive or to reproduce than the others. The new population is produced spinning the roulette-wheel N times, where N represents the total size of the population. This favors individuals with a higher fitness, while leaving a chance for the worst individuals to take part in the reproduction process, hence keeping diversity through the generations. Individuals with an equal rank get the same fitness value and have thus the same probability to survive or to reproduce.

Once the parents have been selected with this method, the offspring can be generated. The offsprings’ genes can be computed using the values of the parents’ genes (Fig. 4). To prescribe the properties of the offsprings, we use randomly one of the three following methods, with probabilities given in Table 1:

Survival: Only one individual is selected and the offspring will be the same as this parent without any change.

Average: Two parents are chosen and the genes of the offspring are the average of the genes of the two parents.

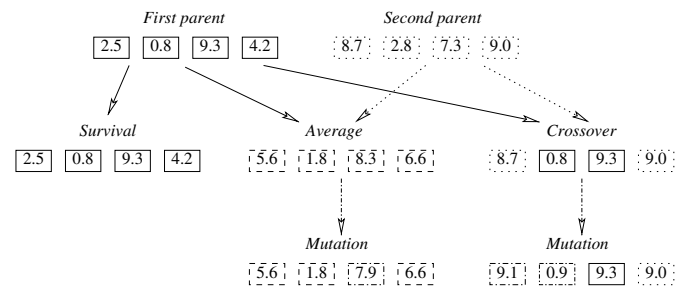


Fig. 4. Principle of the Genetic Algorithms after selection showing survival, average, crossover and mutation. The individuals have four genes, all of them are between 0 and 10.

Table 1
Parameters of the Genetic Algorithm

Parameter	Value
Population size, N	30
Generations	20
Survival probability	0.5
Average probability	0.333
Crossover probability	0.167
Mutation probability	1.0
Mutation magnitude	30% ^a (i.e. $\pm 15\%$)

^a This value is multiplied by 0.8 at each generation. For example the mutation magnitude is 4% ($\pm 2\%$) after 10 generations or 0.43% ($\pm 0.21\%$) after 20 generations. Mutation magnitude must be decreased during the optimization process to stabilize the population.

Crossover: The crossover can be used to increase the diversity among the population. In our problem, the genes are the parameters of the blade profile and these floating-point numbers are selected from either one of the parents and mixed randomly during the crossover process. In this way, randomly selected genes from both parents will be kept in the future generations by being associated with the corresponding offspring.

To introduce diversity, the offsprings further go through a mutation step. A mutation operator is needed because important genetic information may occasionally be lost or missing. In the present case, mutation is a key search operator for the domain exploration so that the mutation probability must be close to 1. A mutation probability of 1 means that all individual genes obtained by averaging or crossover will be modified by mutation. This mutation is performed after defining the offsprings, to randomly modify the offsprings’ genes. Fig. 4 represents a simple example illustrating the procedure.

Multi-objective methods attempt to localize the Pareto front, which is the set of all non-dominated configurations according to the definition given above. Thus, the multi-objective optimization problem aims at finding a discrete approximation of the Pareto front (sometimes also denoted Pareto Optimal Frontier, POF) which is the set of all

non-dominated parameters. As will be shown later, the POF is clearly visible as soon as enough non-dominated configurations have been identified and plotted.

All parameters of the GA procedure used in this work are listed in Table 1. Parameters usually chosen in the literature as well as further parameter sets have been tested extensively in previous works [31], showing that the values retained in Table 1 are more appropriate for the problem considered here.

4. Numerical solution

The set of coupled numerical tools used to solve the multi-objective optimization problem are now described and schematically shown in Fig. 5.

4.1. The OPAL (optimization algorithms) package

OPAL [31] is an object-oriented C++ code for Unix/Linux systems, using a Tcl-script interpreter and optionally a Tk-based [32] graphical user interface. A Tcl-script is used for coupling OPAL with other computer codes, and is used in our case to call a C interfacing program responsible for the evaluation of the objective functions.

4.2. Evaluation of the objectives

In the present case, this evaluation relies on the commercial software Gambit [33] for geometry and mesh generation, and Fluent [26] for solving the flow and energy equations. Therefore, the evaluation of an individual set of parameters requires four steps:

- (1) the generation of the profile contour (blades) from the design variables;
- (2) the generation of an appropriate mesh for the obtained geometry;
- (3) the CFD simulation, i.e. the solution of the governing coupled equations for the flow variable and the energy on the mesh generated in the previous step;
- (4) the post-processing of the obtained results to extract the values of the objective functions for these specific design variables.

Steps 1 and 2 are performed using the commercial software Gambit 2.1 [33], step 3 using the CFD code Fluent 6.1 [26] and step 4 takes place in the in-house interfacing code.

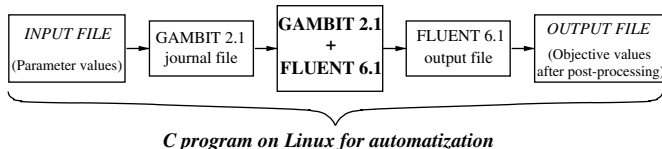


Fig. 5. Flow chart showing the numerical solution procedure.

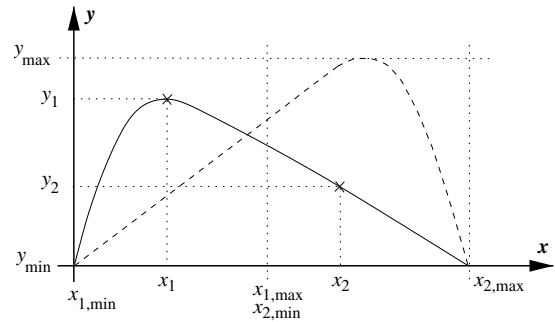


Fig. 6. Basic blade shape described using four design parameters x_1 , x_2 , y_1 and y_2 .

4.2.1. Blade shape (step 1)

Two possible blade shapes are shown in Fig. 6. The points $(x_{1,min}, y_{min})$ and $(x_{2,max}, y_{min})$ are always fixed. The geometrical constraints are prescribed in terms of lower and upper limits on the parameters presented in Fig. 6.

The shape of the blade geometry is defined here using non-uniform rational basic splines (NURBS) [34], where four independent parameters describe half of the blade shape. These four parameters correspond to the two-dimensional Cartesian coordinates of two points, (x_1, y_1) , (x_2, y_2) . These points together with the two fixed points at the extremity completely and uniquely define the NURBS curve in the x - y plane. NURBS-based curves of degree $n = 3$ are proposed as a standard in the GAMBIT software [33]. Such curves consist in a piecewise rational polynomial function of degree n , wherein the numerator and denominator are both non-periodic B-splines of degree n . Natural boundary conditions are applied at the endpoint vertices, using the fact that the second derivative of the NURBS curve is zero. The blade geometry, including its derivative, is fully continuous between the points $]x_{1,min}, x_{2,max}[$, but no further constraints were added to request a differentiable profile at the end points $x_{1,min}$ and $x_{2,max}$, though this would be of course possible. All shapes presented below could nevertheless be machined without difficulty. A further refinement with a finer description of the blade would be possible using more such control points. Note however that increasing the number of points will lead to a much higher number of possible geometries. In that case, further geometrical constraints will in general be necessary to allow the optimizer to deal only with acceptable shapes, that can really be machined at an acceptable cost considering existing techniques and materials.

4.2.2. Mesh generation (step 2)

After having defined the geometry, the mesh is produced in an automatic manner using the commercial software Gambit 2.1 [33]. This is easily done by modifying the journal file containing the important geometrical parameters as variables. Knowing (x_1, y_1) , (x_2, y_2) is sufficient to fully define the geometry and therefore to generate the mesh in an automatic manner and export it to fluent. The script

checks that mesh generation has been successful before going on with the CFD computation, which has always been the case for this simple geometry but might pose a problem for complex three-dimensional configurations. A typical example of the resulting grid is shown in Fig. 2. The basic blade shape is duplicated four times and the internal fluid region is meshed using quadrilateral cell elements using the “pave” algorithm of Gambit. This automatic mesh generation has worked in all the cases without errors, non-feasible cases have not been found. The computational nodes are uniformly spaced along the boundaries. The typical number of computational cells in a mesh lies between 4000 and 5000. A systematic grid-independence study has been performed, as presented in the next subsection.

4.2.3. CFD simulation (step 3)

The in-house interfacing code now starts Fluent [26] in an automatic manner, using again a so-called journal file. Only the geometry and the mesh change between two CFD computations.

The inlet (left side in Fig. 7) boundary is considered as a velocity inlet with imposed conditions for the velocity, set to $v_{\text{inlet}} = 0.5$ m/s, and the temperature $T_{\text{inlet}} = 293$ K. Wall boundary conditions with a constant temperature $T_{\text{wall}} = 353$ K are prescribed on all four blade surfaces. Symmetry conditions are applied in between the blades on the top and bottom boundaries. On the right, a pressure outlet condition relaxing to atmospheric pressure is imposed.

The discretized governing equations are solved iteratively in a segregated manner using a finite-volume description. To improve the accuracy of second-order discretization is systematically used for all variables, along with a double-precision computation. The normalized residuals are computed at every iteration by Fluent. As soon as all of these residuals fall below a prescribed value, convergence is reached. In our case the fixed prescribed value is 10^{-4} for the flow equations and 10^{-6} for the temperature equation, providing a sufficient accuracy for an acceptable CPU time. This point has been checked for one of the solutions identified as optimal, by further decreasing these thresholds. A

completely negligible influence has been observed for the two objective variables.

In the same way, a systematic grid-independence study has been carried out for this selected non-dominated case. By refining several times the grid in a uniform manner the relative pressure and temperature differences do not change by more than 0.8% (0.0026 Pa) respectively 0.065% (0.20 K), demonstrating that the initial grid is sufficient to obtain quantitative estimations.

The velocity–pressure coupling is treated with the standard SIMPLE pressure-correction method. In most cases the convergence is achieved in 100–200 iteration steps. If the convergence is not reached within 500 iteration steps, the simulation is considered as not converging and is fully dismissed. This has been observed for a few evaluations, far from the POF.

4.2.4. Post-processing (step 4)

After convergence, the temperature difference between the inlet (uniform constant value) and the averaged value along the outlet is computed. The pressure difference between the inlet and outlet averaged pressure values is also computed. These two differences are the two objectives of the optimization problem. The resulting temperature and pressure fields of one of the optimum solutions are presented as an example in Figs. 7 and 8.

4.3. Parallelization

As mentioned above, an important issue related with GAs is the high computational effort needed to perform the necessary evaluations of the objectives associated with the population. The evaluation of a large number of individuals for a large number of generations can lead to unaffordable CPU times in practical engineering cases. On the other hand, the GAs have the advantage to be easily and efficiently parallelizable. At each generation, all N individuals can be evaluated independently on different processors, since the central algorithm only needs the values of the objectives to iterate. In the current case, the parallelization has been implemented using the C interfacing program (see Fig. 5) responsible for the evaluation of the $N_{\text{obj}} = 2$

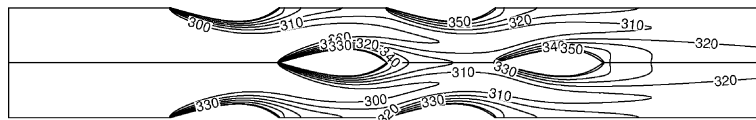


Fig. 7. A typical CFD result, showing the obtained temperature field in Kelvin of one of the optimum solutions.

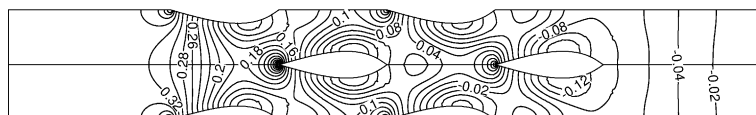


Fig. 8. A typical CFD result, showing the obtained relative pressure field in Pascal of one of the optimum solutions (same solution as in Fig. 7).

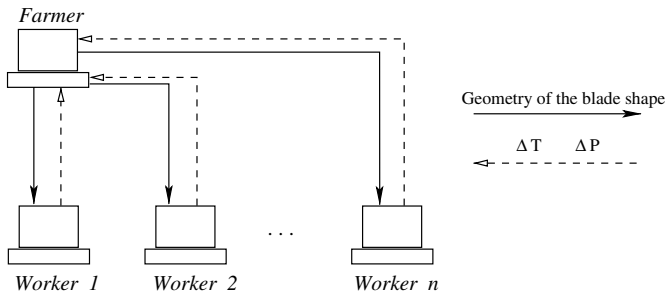


Fig. 9. Schematic flow chart showing the multi-objective CFD optimization problem running on a multi-node PC cluster.

objective values associated with the N configurations. A farmer/worker paradigm has been retained [18], using the free MPICH implementation [35] of the message passing interface (MPI) [36] C communication routines. All the evaluations are carried out on a Linux PC cluster running under Red Hat 9. Fig. 9 shows a schematic description of this multi-processor optimization procedure.

Typical results leading to Pareto fronts can be extracted from these calculations and are presented and discussed in the next section.

5. Computational results

5.1. Pareto fronts

After several tests, 15 PCs were employed for the evaluations and GA optimization was carried out using 30 individuals at every generation. The parents in the first generation are generated using the quasi-random method proposed by Sobol' [37] and modified by Antonov and Saleev [38]. The non-dominated individuals at each generation belong to the elite group. At each iteration 15 individuals can survive for the next generation and 15 new offsprings are added, 10 using averaging and five using crossover, leading again to 30 individuals. Since there are 30 parents and only 15 survive for the next generation, the 15 others disappear from the reproduction cycle but are kept in the elite if they are non-dominated. Nevertheless, in this case, they cannot generate offsprings anymore. If more than 15 non-dominated individuals exist in the present population, the integration within the next generation is again based on the roulette-wheel method described in Section 3.2.

The resulting temperature and pressure differences for all points evaluated over 20 generations are plotted in Fig. 10. Note that 3% of the points lie outside of the figure limits and are not shown here for clarity reasons. The two objectives, temperature and pressure difference, are plotted on the x and y axes, respectively. The remaining points already generate a clearly visible POF in the lower part of the figure.

In Fig. 11 we illustrate again the concept of the Pareto front, which is the boundary between infeasible configurations and possible, but non-optimal solutions.

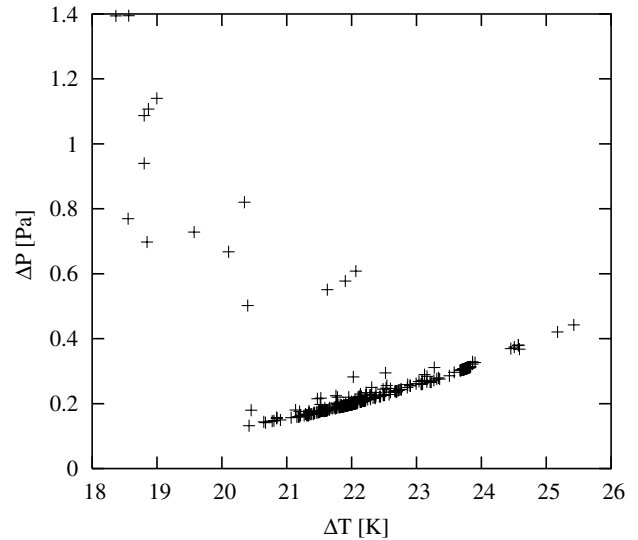


Fig. 10. Results of all evaluations during 20 generations of the GA. A clear approximation of the Pareto front is visible on the lower part of the figure.

Six different optimal blade profiles obtained by the optimization procedure are presented in Fig. 12. The flow direction is from left to right. The first two figures correspond to a low-pressure loss, but the average temperature difference between the inlet and the outlet is small. Higher temperature differences can be achieved using the blades shown for example in Figs. 12(e) and (f), but in this case the pressure losses increase. It is interesting to note that the optimization procedure retains blades turned towards the inflow as well as towards the outflow along the POF.

5.2. GA parameters

The objective values obtained from the simulations are shown as a function of the number of generations in Fig. 13, in which only the non-dominated configurations are represented. We observe that the POF can be recognized very early, i.e. already after 2 or 3 generations. Nevertheless, the quality increases with the number of generations and the POF becomes more accurate. There are two reasons for this progressive improvement: first, we have more and more individuals in the elite group. Secondly, OPAL favors individuals that are very different. For example, if we have five non-dominated individuals with the same fitness value in the parent's group, with four of them very close to each other and the last one quite different, this last individual will be favored for reproduction to enhance diversity, which improves the spatial extension of the POF.

5.3. Speed-up obtained through parallelization

In order to reduce the waiting time for the user, the process has been parallelized and carried out on a multi-node

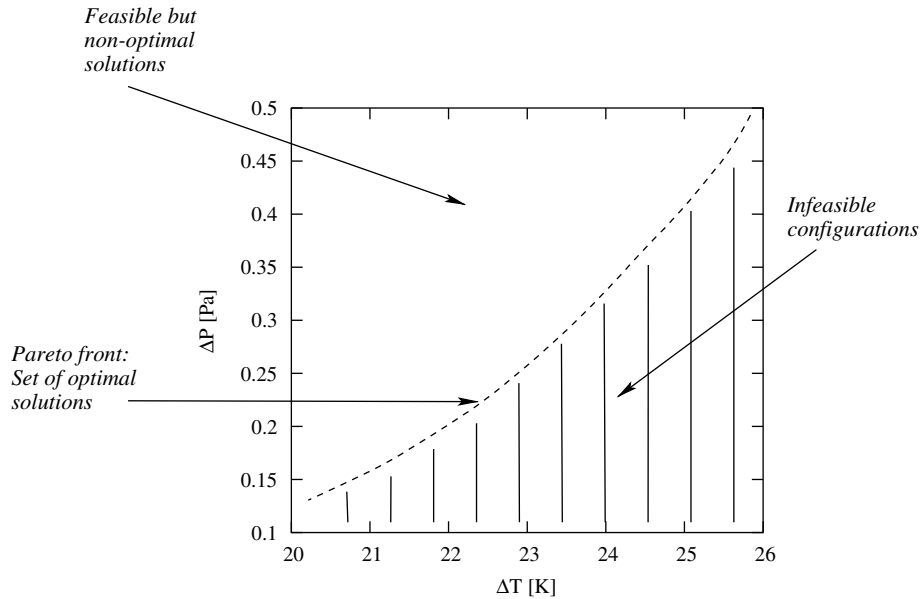


Fig. 11. Pareto optimal front (POF) obtained after 20 generations with 30 individuals. Only the boundary curve of the non-dominated configurations is represented here (see also Fig. 10).

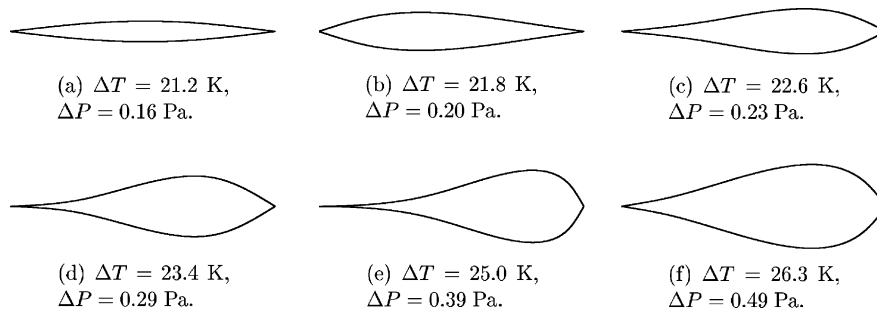


Fig. 12. Example of the resulting blade profiles for six individuals belonging to the POF. The flow direction is from left to right.

Linux PC cluster with 15 worker PCs. Each node is a 2.6 GHz/2 GB-RAM Pentium-IV Linux PC. The communications are performed via a Fast Ethernet (1 Gb/s) network connection. Table 2 shows the resulting CPU times needed for the evaluation of 20 generations consisting of 30 individuals, performed with an increasing number of processors on the PC cluster. The speed-up is defined here as the ratio between the wall-clock time obtained when using N_{proc} processors compared to the one needed with a single processor. The theoretical optimal value of the speed-up using N_{proc} processors is N_{proc} . In practical cases, the communication times between processors reduce the speed-up value below the theoretical maximum.

In the present case, the obtained parallel speed-up is nearly optimal. This is not really a surprise, since the quantity of information transferred between the so-called farmer and workers is very small: four real parameters in one direction, two in the reverse direction. The communication times are therefore almost negligible compared to the CPU times required for the evaluation of the objectives on each processor. Deviation from the optimal speed-up

are thus mainly due to boundary effects at the end of each optimization iteration, when some worker PCs become inactive for a short time, waiting for the next iteration to start.

6. Summary

In this study, a genetic algorithm has been applied to a multi-objective shape design optimization problem concerning a heat exchanger configuration close to practical applications. The characteristic Pareto front associated with this problem has been obtained within a very reasonable computational time.

This model problem is based on a description of the blade geometry using only four parameters. This is clearly a very simple description that could be refined. Computing times can be further reduced by using parallelization on up to 15 nodes if needed, as demonstrated in this paper. More complex, practical industrial cases are solvable using on one side appropriate modeling and simplification of the problem and on the other side parallelization.

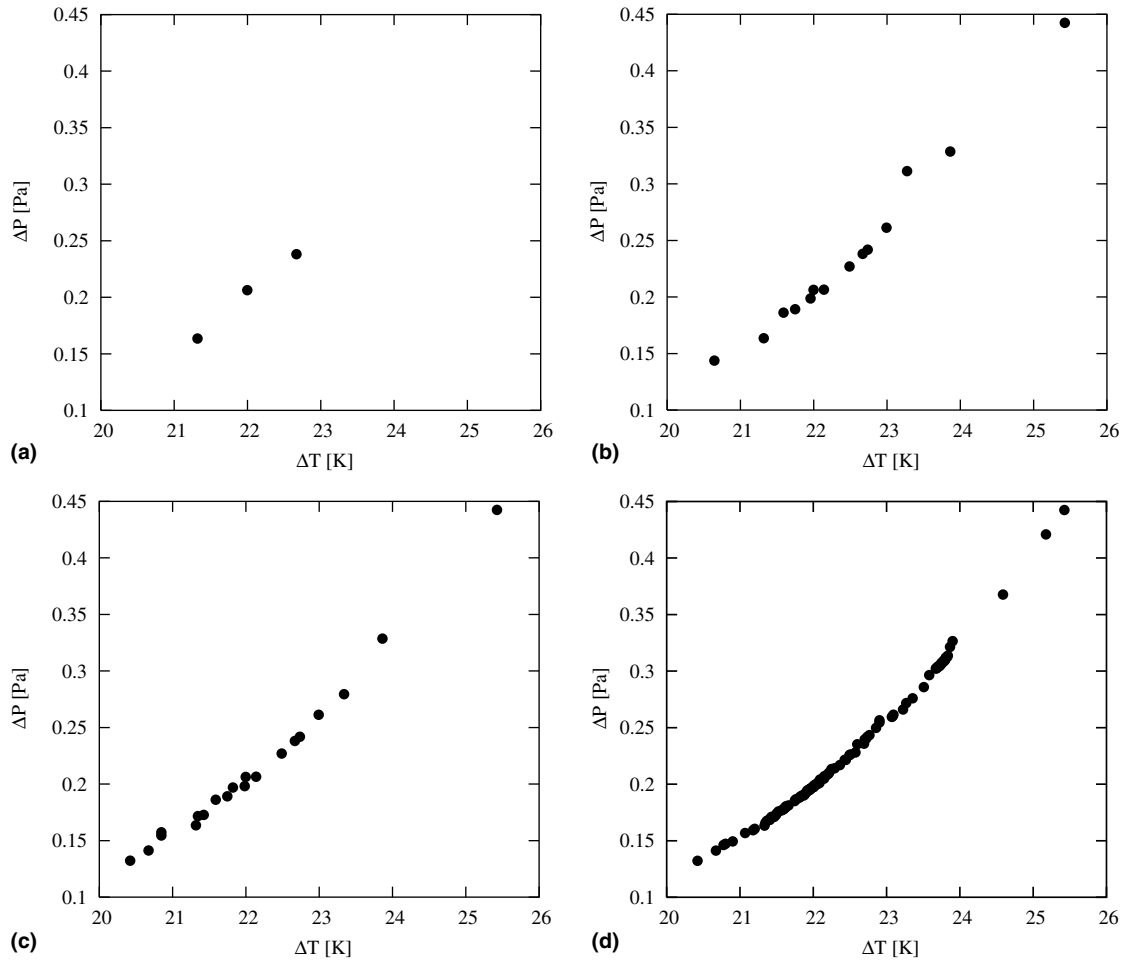


Fig. 13. The non-dominated individuals (elite). (a) The non-dominated individuals after one generation. (b) The non-dominated individuals after two generations. (c) The non-dominated individuals after three generations. (d) The non-dominated individuals after 20 generations.

Table 2
Speed-up obtained with the parallel optimization method

Number of processors	Wall-clock time	Speed-up
1	129 min 10 s	1
3	44 min 6 s	2.93
5	27 min 17 s	4.73
8	18 min 52 s	6.85
15	10 min 55 s	11.83

Further investigations are presently conducted to decrease the needed CPU time even further and therefore to have easily access to more realistic configurations, described with more refined models using a higher number of parameters. Two possible improvements are currently tested. On one side, speed-up can be achieved in the optimization algorithm itself. For example, some results (see e.g. [39,40]) indicate that the hybridizing of the GA with classical gradient-like methods is an interesting way to follow. On the other side, new methods can be used for the evaluation of the objective values, such as the coupling of the GA with an artificial neural network [3,41] or the application of statistical methods such as those based on the response surface concept.

Acknowledgement

The development of OPAL would not have been possible without the financial support of the CETIAT (Coordinator: G. Perrin) and of ADEME during the Ph.D. of Romain Baron. This research project has been initially started at the EM2C Laboratory (École Centrale Paris, France).

References

- [1] I.D. Falco, An introduction to Evolutionary Algorithms and their application to the aerofoil design problem – Part I: the Algorithms, von Kármán Lecture Series on Fluid Dynamics, Bruxelles, Belgium, April 1997.
- [2] R. Mäkinen, P. Neittaanmäki, J. Périaux, J. Toivanen, A genetic algorithm for multiobjective design optimization in aerodynamics and electromagnetics, in: K.D. Papailiou, D. Tsahalis, J. Périaux, D. Knörzer (Eds.), Computational Fluid Dynamics '98, Proceedings of the ECCOMAS 98 Conference, vol. 2, Wiley, Athens, Greece, 1998, pp. 418–422.
- [3] S.Y. Han, J.S. Maeng, Shape optimization of cut-off in a multi-blade fan/scroll system using neural network, Int. J. Heat Mass Transfer 46 (15) (2003) 2833–2839.

- [4] K.S. Lee, W.S. Kim, J.M. Si, Optimal shape and arrangement of staggered pins in the channel of a plate heat exchanger, *Int. J. Heat Mass Transfer* 44 (17) (2001) 3223–3231.
- [5] J.I. Madsen, W. Shyy, R.T. Haftka, Response surface techniques for diffuser shape optimization, *AIAA J.* 38 (9) (2000) 1512–1518.
- [6] G. Fabbri, Heat transfer optimization in corrugated wall channels, *Int. J. Heat Mass Transfer* 43 (23) (2000) 4299–4310.
- [7] C.H. Cheng, M.H. Chang, Shape design for a cylinder with uniform temperature distribution on the outer surface by inverse heat transfer method, *Int. J. Heat Mass Transfer* 46 (1) (2003) 101–111.
- [8] G. Fabbri, Heat transfer optimization in internally finned tubes under laminar flow conditions, *Int. J. Heat Mass Transfer* 41 (10) (1998) 1243–1253.
- [9] D. Balagangadhar, S. Roy, Design sensitivity analysis of steady fluid–thermal systems, *Comput. Methods Appl. Mech. Eng.* 190 (2001) 5465–5479.
- [10] J. Bonjour, L.A.O. Rocha, A. Bejan, F. Meunier, Dendritic fins optimization for a coaxial two-stream heat exchanger, *Int. J. Heat Mass Transfer* 47 (1) (2004) 111–124.
- [11] G. Fabbri, Effect of viscous dissipation on the optimization of the heat transfer in internally finned tubes, *Int. J. Heat Mass Transfer* 47 (14–16) (2004) 3003–3015.
- [12] S. Tiwari, D. Maurya, G. Biswas, V. Eswaran, Heat transfer enhancement in cross-flow heat exchangers using oval tubes and multiple delta winglets, *Int. J. Heat Mass Transfer* 46 (15) (2003) 2841–2856.
- [13] R.S. Matos, J.V.C. Vargas, T.A. Laursen, A. Bejan, Optimally staggered finned circular and elliptic tubes in forced convection, *Int. J. Heat Mass Transfer* 47 (6–7) (2004) 1347–1359.
- [14] F. Kock, H. Herwig, Entropy production calculation for turbulent shear flows and their implementation in CFD codes, *Int. J. Heat Fluid Flow* 26 (2005) 672–680.
- [15] A. Bejan, *Entropy Generation Minimization*, CRC Press, Boca Raton, FL, 1996.
- [16] J.A. Visser, D.J. de Kock, Optimization of heat sink mass using the DYNAMIC-Q numerical optimization method, *Commun. Numer. Methods Eng.* 18 (10) (2002) 721–727.
- [17] T. Okabe, K. Foli, M. Olhofer, Y. Jin, B. Sendhoff, Comparative studies on micro heat exchanger optimization, *Proc. IEEE Congress Evolution. Comput.* 1 (2003) 647–654.
- [18] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 1989.
- [19] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, Wiley, London, 2001.
- [20] J. Lee, P. Hajela, Parallel genetic algorithm implementation in multidisciplinary rotor blade design, *J. Aircraft* 33 (5) (1996) 962–969.
- [21] R.A.E. Mäkinen, J. Périaux, J. Toivanen, Multidisciplinary shape optimization in aerodynamics and electromagnetics using genetic algorithms, *Int. J. Numer. Methods Fluids* 30 (2) (1999) 149–159.
- [22] S. Obayashi, T. Tsukahara, T. Nakamura, Multiobjective genetic algorithm applied to aerodynamic design of cascade airfoils, *IEEE Trans. Ind. Electron.* 47 (1) (2000) 211–216.
- [23] N. Ali, K. Behdinan, Optimal geometrical design of aircraft using genetic algorithms, *Trans. Canad. Soc. Mech. Eng.* 26 (4) (2003) 373–388.
- [24] V. Nejati, K. Matsuuchi, Aerodynamics design and genetic algorithms for optimization of airship bodies, *JSME Int. J. Ser. B – Fluids Therm. Eng.* 46 (4) (2003) 610–617.
- [25] F. Muyl, L. Dumas, V. Herbert, Hybrid method for aerodynamic shape optimization in automotive industry, *Comput. Fluids* 33 (5–6) (2004) 849–858.
- [26] Fluent Inc., *FLUENT 6.1 User's Guide*, Lebanon, New Hampshire, 2003.
- [27] N. Srinivas, K. Deb, Multiobjective optimization using non-dominated sorting in genetic algorithm, *Evolution. Comput.* 2 (3) (1995) 221–248.
- [28] J.A. Nelder, R. Mead, A simplex method for function minimization, *Comput. J.* 7 (1965) 308–313.
- [29] C.M. Fonseca, P.J. Fleming, Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, in: S. Forest (Ed.), *Genetic Algorithms: Proceedings of the Fifth International Conference*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 416–423.
- [30] Z. Michalewicz, *Genetic Algorithms+Data Structures = Evolution Programs*, Springer-Verlag, Berlin, Heidelberg, New York, 1996.
- [31] R. Baron, *Calcul et optimisation de brûleurs laminaires industriels*, Ph.D. Thesis, École Centrale Paris, 2002–37, 2002.
- [32] Tcl developer site. <<http://www.tcl.tk/>>.
- [33] Fluent Inc., *GAMBIT 2.1 User's Guide*, Lebanon, New Hampshire, 2003.
- [34] D.F. Rogers, *An Introduction to NURBS: with Historical Perspective*, Morgan Kaufmann Publishers, San Francisco, 2001.
- [35] MPICH – A portable implementation of MPI. Available from: <<http://www-unix.mcs.anl.gov/mpi/mpich/>>.
- [36] The MPI standard. Available from: <<http://www-unix.mcs.anl.gov/mpi/>>.
- [37] I.M. Sobol', Distribution of points in a cube and approximate evaluation of integrals, *USSR Comput. Math. Math. Phys.* 7 (1967) 86–112.
- [38] I.A. Antonov, V.M. Saleev, An economic method of computing LP-sequences, *USSR Comput. Math. Math. Phys.* 19 (1979) 252–256.
- [39] D.T. Pham, G. Jin, Genetic algorithm using gradient-like reproduction operator, *Electron. Lett.* 31 (18) (1995) 1558–1559.
- [40] K. Rasheed, Guided crossover: a new operator for genetic algorithm based optimization, in: P.J. Angeline, Z. Michalewicz, M. Schoenauer, X. Yao, A. Zalzal (Eds.), *Proceedings of the Congress on Evolutionary Computation*, vol. 2, IEEE Press, 1999, pp. 1535–1541.
- [41] E. Kazancioglu, G.W.J. Ko, S. Bohac, S.J. Hu, D. Assanis, K. Saitou, Robust optimization of an automobile valvetrain using a multiobjective genetic algorithm, in: *Proceedings of DETC'03 ASME 2003 Design Engineering Technical Conferences Chicago DETC2003/DAC-48714*.