Proceedings of ICFDP7:
SEVENTH INTERNATIONAL CONFERENCE OF FLUID DYNAMICS AND PROPULSION
Dec. 18-20, 2001, Cairo, EGYPT

ICFDP7-2001055

A PARALLEL GENETIC ALGORITHM APPLIED TO THE DESIGN OF BLADE PROFILES FOR CENTRIFUGAL PUMP IMPELLERS

Waleed A. Wahba
Rockets Department
Military Technical College
Cairo, Egypt

Antonios Tourlidakis
School of Engineering
Cranfield University
Cranfield, Bedford, UK

ABSTRACT

This paper presents a design method for blade profiles of centrifugal pump impellers based on a parallel optimisation algorithm in combination with Computational Fluid Dynamics (CFD). An optimisation library (GAlib), based on a genetic algorithm (GA), has been modified in order to be able to use any number of processors using a master/slave method where the master stores the population, executes GA operations, and distributes individuals to the slaves. The slaves only evaluate the fitness of the individuals and at the same time the master evaluates one of the fitness as well. GA controls the evolution of a population of profiles towards an optimum design. The fitness value of each population element is evaluated using a CFD flow solver (Mac_LNS) based on finite-difference discretisation of the incompressible, Navier-Stokes equations on structured polar-coordinate meshes. A number of design examples have been investigated and the behaviour of the genetic algorithm has been tested using different kinds of objective functions. In addition, the algorithm was tested with a multi-objective function using weighted sum method. Bézier curves were selected to represent the impeller profile. A symmetric profile, identical profile for the pressure side and suction side, was used as basic shape to generate the population elements. The obtained results show that the parallised genetic algorithm is capable of achieving satisfactory designs of centrifugal impeller blade profiles effectively, faster and with a minimum amount of user expertise.

INTRODUCTION

The purpose of the centrifugal machine (pump, or compressor) is to add energy to or produce pressure rise in the fluid, which flows through it. The rotating impeller is the most important, the only rotating element of the pump. The diffuser, following the impeller, can transform kinetic energy into pressure energy, which is equivalent to approximately 20% to 40% of the total work input, but cannot increase the total energy of the fluid. The shape of the blades and the resulting flow pattern in the impeller determine how much energy is transferred by a given size impeller and how efficiently it operates.
In today's competitive world, pump designers are tempted to just quickly copy existing pumps, or fall back on empirical design formulas established by statistical surveys of existing pumps. Great problems arise, however, when the designs do not perform as expected. Without the deeper understanding of the flow pattern in pumps, the reasons for malfunctioning can not be explained. Statistically average designs may be adequate but will not provide a competitive edge. Today's pump designers need computer codes that have been built on a deep understanding of the flow patterns inside the pumps. Specially, present computers have higher speed and memory than those where empirical formulas were implemented.

During the last two decades significant progress has been accomplished using CFD. CFD has become such an effective tool that plays very powerful role to achieve design goals more rapidly and cost effectively [1]. Moreover, up to now, several factors has limited the use of CFD alone in the industrial design environment. However, progress has been reported on the use of CFD in design by linking it with CAD [2] or optimisation algorithms [3-5].

Numerical methodologies for solving the problem of direct shape design are classified into two main classes: direct (optimisation) and inverse design methods. The inverse design methods require an 'a priori' definition of an "optimal" target distribution (pressure or velocity) by the designer. This obviously demands a close understanding of the flow phenomena considered, and furthermore involves a problem

regarding the existence of solutions. For these reasons, it is hopeless to apply inverse design methods for general purposes, while the methods may prove very efficient for specific problems, which are well understood, such as for the design of pump impellers, [6]. Dulikravich [7] published a survey of inverse design methods. In the field of turbomachinery blade design, von Karman Institute (VKI) used inverse design methods with many flow solvers like, potential flow, Euler, and Navier-Stokes, [8]. On the other hand, in the optimisation (direct) methodology, the design problem becomes a minimisation or maximisation problem of an objective function subjected to constraints on the geometric and fluid properties. Optimisation methods assist the designer in locating the min/max of the objective function while satisfying the constraints. Direct methods can be distinguished into two categories: global methods and local methods. Global methods, such as those based on GAs [9], are aimed at obtaining the global optimum, these methods are most useful for cases in which multiple minima or maxima are present in the design space. Local methods use the information on the gradient of the objective for locating the optimum in a localised area. More details about gradient optimisation, [10].

Performance optimisation of centrifugal impeller profiles and the automation of design tasks is an active research field, because there are many objectives to optimise like maximum head and uniform flow (minimum vortices, minimum losses, zero reverse flow etc). Traditionally, there were three principal methods for determining the shape of blades circular arc method, point by point method, and the conformal representation method, [11]. These methods require extensive expertise and the accomplishment of the final design required a large number of prototype construction and testing.

This paper is based on the novel optimisation procedure that has been presented by Wahba and Tourlidakis [12] by linking an optimisation algorithm GAlib, [13], with the CFD code Mac_LNS [14,15], where the designer can obtain his own impeller design by specifying his requirements (objectives) from this impeller with reduced effort, experience and experimental work. The present paper presents the method used to parallelise this optimisation procedure in order to reduce the procedure's time.

SERIAL GAs

The simplest form of a genetic algorithm is the serial GA. Here, there is one population and only one processor to perform the algorithm, as seen in figure 1.
The contour of the blade profile to be optimised is defined (assuming zero thickness) by seven difference angles $d\phi_i$ starting from the leading edge to the trailing edge. The inlet and outlet radii (R_1, R_2), and impeller speed are kept fixed during the optimisation process. While the difference angles $d\phi_i$ are perturbed to change the blade geometrical shape using De Casteljau's algorithm [16].

```
- Initial random population
- Evaluate(calculate fitness of individuals)
- While(generation < MAX_GEN_NO)
        select(select N members of population)
        crossover
        mutate
        evaluate
        generation = generation + 1
- EndWhile
- Return (individual with greatest fitness)
```

Figure 1, Code for a serial genetic algorithm.

PARALLEL GAs

There are many reasons for parallelising the genetic algorithm. The most obvious is of course speed. GAs are computationally expensive compared to most of the more deterministic forms of search and optimisation, especially if the objective function was obtained through CFD, where CFD takes a long time to converge. Additionally, the probabilistic approach taken by a GA as it must search a larger area (more objective function calls) than a deterministic algorithm. These factors make GAs ideal for parallelisation. In some cases, the GA can be parallelised to the point that each individual chromosome is attributed to its own processor to perform necessary computations. This essentially reduces the running time for each generation to the amount of time required for performing the genetic operations on just one individual. Speed is not the only reason for parallelising a genetic algorithm. Some implementations of parallel GAs have a significantly higher cost than serial GAs, instead they increase the chance of finding the optimal solution.

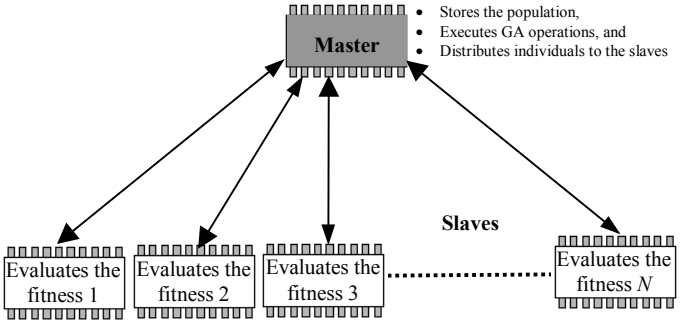

Figure 2, A schematic of a master/slave parallel GA.

In this work GAlib has been modified to use any number of processors using a master/slave method, figure 2, where the master stores the population, executes GA operations, and distributes individuals to the slaves. The slaves only evaluate the fitness of the individuals and during this time the master evaluates one of the fitness as well, as seen in figure 3.

Message Passing Interface (MPI) is used in this parallelisation. MPI efficiently manages message buffers by sending and receiving directly from the user data structures not by buffers within the communication library and therefore buffering may

be totally avoided. A program written for MPI is completely portable, and it is easy to recompile and run on any computer platform. The author's personal view is that, MPI is easier to use than that Parallel Verture Machine (PVM). GAlib has available parallel processing class, which is using PVM. The modification is so simple and it only requires to replace the DefaultEvaluator(GAPopulation &) function with a new function called Eva_POP(GAPopulation &). Full listing of the function Eva_POP(GAPopulation &) is presented in [17].

```
Initial random population                          <Master>
Distribute population on slave processes           <Master>
For each master & slave process p
        evaluate each chromosome
Wait until all slaves have returned their results (fitness)
While(generation < MAX_GEN_NO)                     <Master>
    select(select N members of pop)                <Master>
    crossover                                      <Master>
    mutate                                         <Master>
    For each master & slave process p
            evaluate each chromosome
    Wait until all slaves return their results (fitness)
    generation = generation + 1                    <Master>
EndWhile                                           <Master>
Return(individual with greatest fitness)           <Master>
```

Figure 3, Code for a parallel genetic algorithm.

RESULTS

The results can be classified into three groups: parallel optimisation results, design objectives' results, and design using multi-objectives results.

Parallel Optimisation

In order to demonstrate how computationally expensive a serial GA is, one objective for example "min. friction losses" is proposed using serial GA. Table 1 presents some important "optimisation statistics" used to achieve this objective.

Table 1, Min. friction losses optimisation statistics.

30	Population size
100	current generation
1.00226	current convergence
1600	number of selections since initialisation
857	number of crossovers since initialisation
1374	number of mutations since initialisation
1500	number of replacements since initialisation
1275	number of genome evaluations since initialisation
101	number of population evaluations since initialisation
220 seconds	Average time to evaluate one genome for 10000 CFD iterations for computer PIII, 1000 MHz.

As seen in Table 1, each fitness value calculation needs an average of 220 seconds for 10,000 convergence iterations, which is the average number of iterations for the CFD code to converge. Sometimes 30,000 iterations are required depending on the complexity of the blade profile. On the other hand, the objective function has been executed 1,275 times for 101 population evaluations (100 generation as input) for 30 populations. Consequently the average time of 280,500 seconds is required to complete this case using a serial GA, neglecting the optimisation operators (selection, crossovers, mutation and replacement) solution time. On the other hand, in case of parallel optimisation, with 15 processor, 36,507 seconds has been recorded. The next two steps describe how this number of processors has been selected:

- GA starts with randomly initialisation of a population of individuals, followed by individual evaluation for all 30 individuals. This occurs in three steps: 1) calculate first individual using the master processor to sign some commune variables; 2) distribute next 29 individuals to the slaves and master, and collect objective values from slaves in two rounds (2x15 processors). It can be noticed that the number of processors in two rounds are greater than number of remained individuals, so, the last processor will update its vales and sleep during the second time; 3) apply optimisation operators using the master node.
- After the initialisation step, evaluation will continue for 100 generation assumed, for other cases it can be 50, 150 generation or specified convergence criteria, as previously. If the number of individuals is less than or equal to the number of processors, (this is usually the case), one generation acts like one individual evaluation; if not it needs another round, which means two individual evaluations.

The main disadvantage of this method is that if the CFD calculation in one processor is huge, requiring task a large number of iterations to converge, all processors have to wait until this processor completes its task. Nevertheless, parallelisation is a very good approach to perform the calculations more than seven times faster than the serial one.

Design Objectives

There are five classes of design objectives from a mathematical point of view: integral, local, min/max and max/min, and multi-criteria. In this paper min. losses will be presented as an integral criteria. On the other hand the local criteria can be achieved through minimum pressure side PS relative flow velocity to reduce separated jet-wake flow. The other two design objective classes will be discussed in the design using multi-objectives.

Integral criteria

For the impeller design, good impeller performance needs minimising impeller losses. These losses are made up of impeller skin friction, and dynamic head losses. Impeller dynamic losses are entry shock loss and hydraulic loss. Entry shock losses at the design point are small and can be neglected in this study, where optimisation procedure takes place at design conditions. In this work, hydraulic losses have been defined by total pressure losses. Where less change in blade angle along the blade passage means reduced total pressure losses and longer blade length. Impeller skin friction losses are affected by changing blade length and relative velocity.

Although flow patterns in the impeller flow passages are complex, the application of pipe friction formulae offers to date the best way of correlating design parameters with the magnitude of this loss, [18].
Maximum head, minimum friction losses and total pressure losses, as integral objectives, have been applied, only one each time. Figures 3a,b&c present change of head, efficiency, and finally friction and total pressure losses with different mass flow rates for these three integral criteria, respectively.
The use of the first objective "maximum head" offers better head and efficiency characteristics than those provided by the other objectives "minimum friction and total pressure losses", but there are a lot of parameters which lead the designer to avoid this objective as will be seen in figure 4.
The second objective "minimum friction losses" has better characteristics than the one provided by the third objective "minimum total pressure losses", where these better characteristics can be attributed in reduced friction losses which has affected both the head and efficiency characteristics. This excess in friction losses in the third objective "minimum total pressure losses" is due to more blade passage length.
Figure 4 presents the relative flow velocity profile in a blade-to-blade view of the impeller for three differently designed geometries at design point. In cases a and b the blade profile becomes concave in the pressure side just after the leading edge which is unstable (particle moves away from wall) for the reasons of concave curvature and Coriolis effect. A very smooth change of the blade angle appears in the case of minimum total pressure losses (case c), which means long blade profile and relatively large friction losses.
On the other hand, figure 5 presents a different aspect of comparison between these three objective cases, which is the optimiser convergence criterion. Figures 5a&b show the convergence history of the two objective cases max. head and min. total pressure losses, respectively. In the case of max. head objective (case a) convergence required large number of generation if it is compared with the case of min. total pressure losses. This is because there are many conflicting issues will be discussed in the section where multi-objective criteria are used. Otherwise, the two curves are quite reasonable form the convergence point of view.

Local Criteria

Local criteria are used to cover any sub properties of the flow-field or geometry. In case of heavy blade loading, the flow calculation may predict that the velocity slows to zero at the pressure side, as was shown above in figure 6.7 (in case of max. head and min. friction losses), and that flow separation will occur along the pressure side. However, since a separated region cannot continue on the pressure side, the flow switches to a jet-wake flow pattern. From this point of view, the designer can consider jet-wake flow pattern as a local criteria which he/her can manage to minimise its effect. This objective "minimum jet-wake flow pattern" has been proposed in this work, where the difference between the relative velocity on the pressure side and suction side, respectively, has been applied to be minimum after the pressure side relative velocity becomes greater than the suction side one.
Figure 6 shows the pressure and suction sides relative flow velocity for this local criterion as compared with some integral criteria. For the local criteria, minimum jet-wake flow pattern, the area between the PS and SS after 85% of the blade length looks slightly smaller than for the other cases which starts earlier (around 75% to 80% of the blade length).

Multi-Objective Design

The majority of engineering design problems requires the simultaneous optimisation of more than one objective function (multi-objectives). It is unlikely that the different objectives would be optimised by the same alternative parameter choices. Hence, some trade-off between the criteria is needed to ensure a satisfactory design. Weighted sum method is used in this paper, where hierarchical approach using penalty method based on feasibility has been used in [12]. The approach most users of GAs favour to the problem of multi-objective, is to weight and sum the separate fitness values in order to produce just a single fitness value for every individual. Two objectives, total pressure and friction losses (minimum losses) will take place, where a weighting factor, R_1 and R_2 multiply each one, respectively. And as in the penalty function method, users usually have to try different values of penalty parameter, to find what value would be the best. Here also, these two weighting factors will start from equal values (0.5 & 0.5) and change slightly. This change will be in one direction only, increasing the total pressure losses case weighting factor and decreasing the friction losses one. Figure 7 presents the impeller's profile loading for this case. It can be seen that the sum of the weighted objectives method works well for multi-objective functions where some intermediate design cases have been obtained. Also, figure 8 shows the impeller's profiles geometry for these six design cases, where it appears that the decision maker can find a never-ending range and discover more design profiles. All new four design cases provided in this subsection are quite acceptable.

CONCLUSIONS

This work presented a novel way to design the pump impeller blade profile using parallel GAs and CFD. As CFD is used in the current study, results show that it is a quite good step to boost the power of CFD by integration with optimisation tools for an automated design procedure.
The paper works step by step how a number of objectives can be tested and how each objective behaves. Single and multiple objectives are tested. Two different kinds of single objective, integral and local, have been used. Weighted sum method is used to handle the multi-objective criteria. Setting multi-objective criteria is very essential in order to find a good design. It is indicated that parallelisation is a good technique to overcome the time taken by GA and CFD, and by

parallelisation this procedure can be extended to achieve design optimisation using CFD to 3D cases.

ACKNOWLEDGEMENT

The software for this work used the GAlib genetic algorithm package, written by Matthew Wall at the Masschusetts Institute of Technology, MIT.

REFERENCES

[1] Orszag S.A. and Staroselsky I., 2000, "CFD: Progress and Problems," Computer Physics Communications 127, pp. 165-171.

[2] Al-Zubaidy S.N., 1995, "A Proposed Design Package for Centrifugal Impellers," Computers & Structures, Vol. 55, No. 2, pp. 347-356.

[3] Narducci R., Grossman B., Valorani M., Dadone A. and Haftka R.T., 1995, "Optimization Methods for Non-smooth or Noisy Objective Functions in Fluid Design Problems," AIAA Paper 95-1648.

[4] Johansen M.D., et.al., 1997, "Response Surface Approximations for Shape Optimization of a 3D-Manifold," Proceedings of Fourth CFX International Users Conference, Chicago, IL, October 6-10, pp. 222-231.

[5] Madsen J., et.al., 1997, "Response Surface Techniques for Diffuser Shape Optimization", AIAA Paper 97-1801.

[6] Zangeneh M., Goto A. and Takemura T., 1996, "Suppression of Secondary Flows in a Mixed-Flow Pump Impellers by Application of Three-Dimensional Inverse Design Method: Part 1- Design and Numerical Validation", J. Turbomachinery 118, pp. 536-543.

[7] Dulikravich G.S., 1991, "Aerodynamic Shape Design and Optimization," AIAA Paper 91-0476.

[8] Van den Braembussche R.A., Pierret S. and Demeulenaere A., 1998, "Modern Turbo-Machinery Blade Design," VKI /Reprint- 1998/21,

[9] Doorly D.J. and Peirò J., 1997, "Supervised Parallel Genetic Algorithms in Aerodynamic Optimisation," AIAA Paper 97-1852.

[10] Pandya M. and Baysal O., 1997, "Gradient-based Aerodynamic Shape Optimization using Alternating Direction Implicit Method," J. of Aircraft, 34(3), pp. 346-352.

[11] Lazarkiewicz S. and Troskolanski A.T., 1965, *Impeller pumps*, Pergamon Press Ltd, London.

[12] Wahba W.A. and Tourlidakis A., 2001, "A Genetic Algorithm Applied to the Design of Blade Profiles for Centrifugal Pump Impellers", 15th AIAA Computational Fluid Dynamics Conference, 11-14 June, Anaheim, CA, USA.

[13] Wall M., 1996, "GAlib: A C++ Library of Genetic Algorithm Components", version 2.4, Massachusetts Institute of Technology (MIT).

[14] Wahba W.A., Abdallh H. and Allam M., 1998a, "Flow Pattern Solution for a Radial Blade Centrifugal Pump", 7th International Conference on Aerospace Sciences & Aviation Technology, 13 - 15 May, MTC, Cairo, Egypt.

[15] Wahba W.A., Abdallh H. and Allam M., 1998b, "Flow Pattern Solution for a Backward Blade Centrifugal Pump", 7th International Conference on Aerospace Sciences & Aviation Technology, 13-15 May, MTC, Cairo, Egypt.

[16] Qiu-Lin Ding and Davies B.J., 1987, *Surface Engineering Geometry for Computer-Aided Design and Manufacture*, Ellis Horwood series in mechanical engineering, Chichester: Horwood.

[17] Wahba W.A., 2001, "Design Optimisation of Centrifugal Pump Impellers using Parallel Genetic Algorithm", Ph.D. thesis, Cranfield University, Bedford, UK.

[18] Neumann B., 1991, *The Interaction Between Geometry and Performance of a Centrifugal Pump*, Mechanical Engineering Publications Ltd, London.

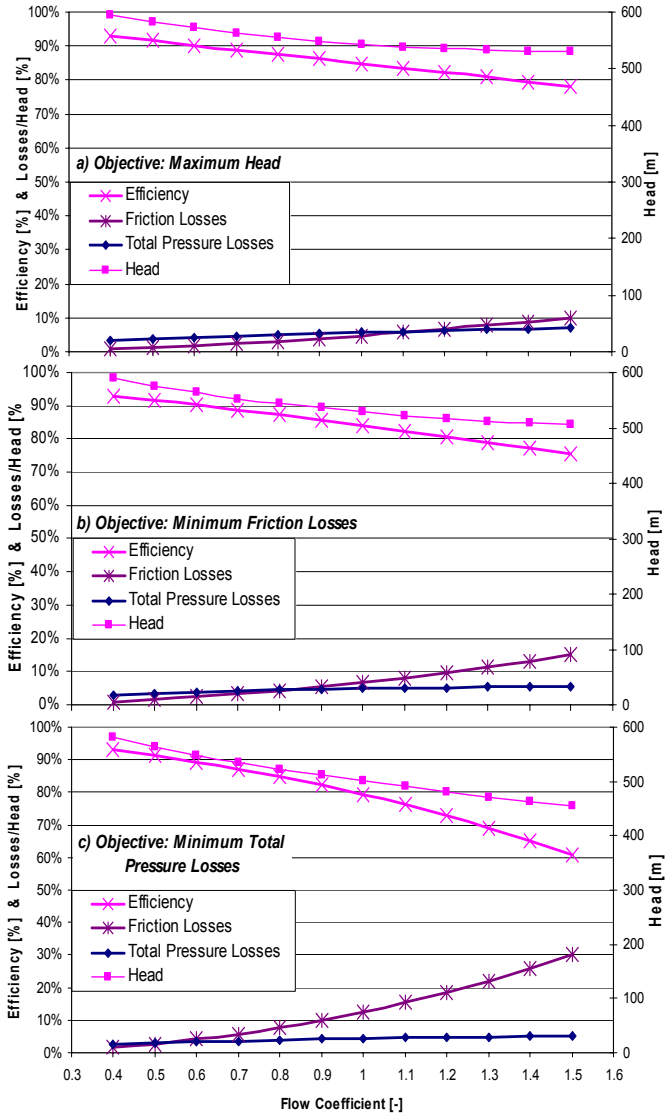

Figure 3, Impeller characteristics with mass flow rat for different integral objectives.

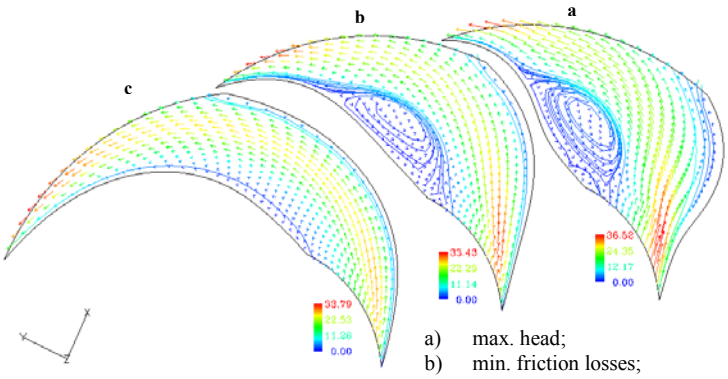

Figure 4, Relative flow velocity in blade-to-blade impeller profiles for three different objective functions.

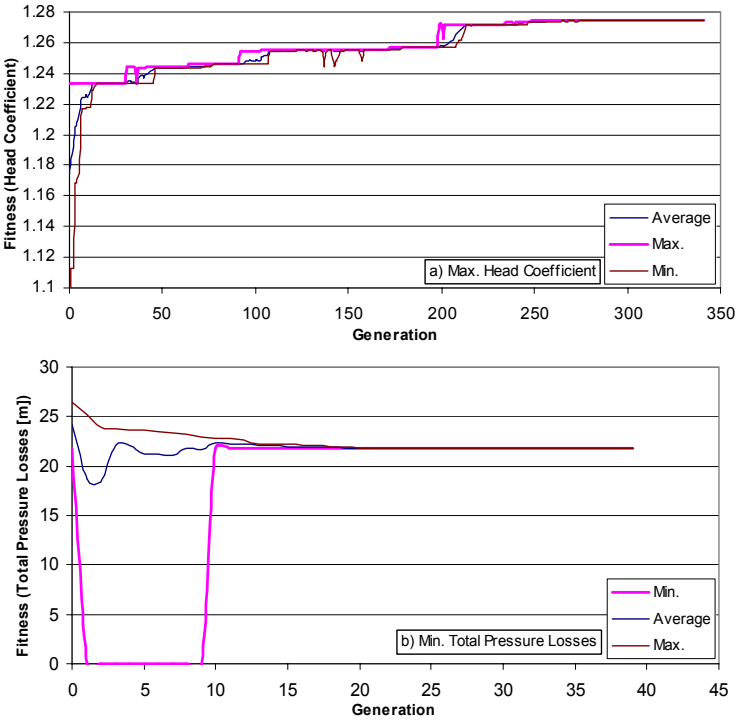

Figure 5, Genetic Algorithm Convergence History.

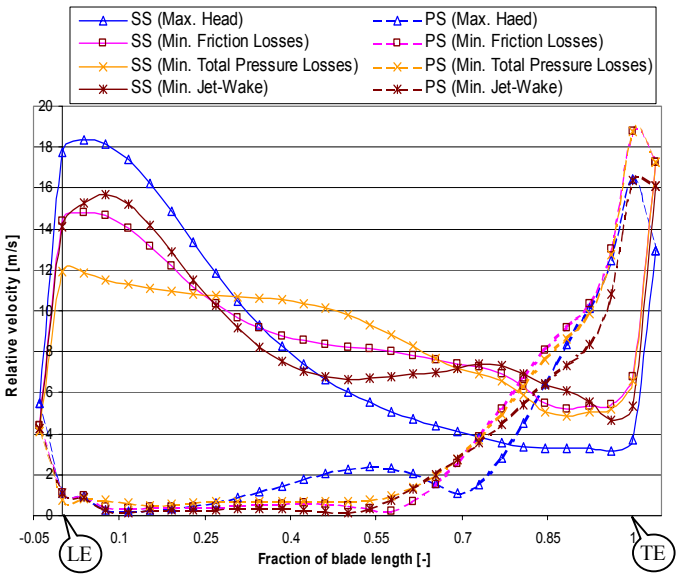

Figure 6, PS and SS relative velocity for three integral and one local objectives at the design point.

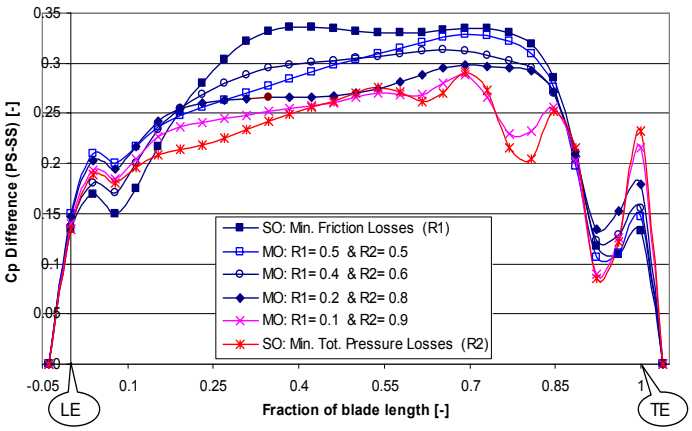

Figure 7, Impeller's profile loading at the design point (Multi-Objectives "weighted sum").

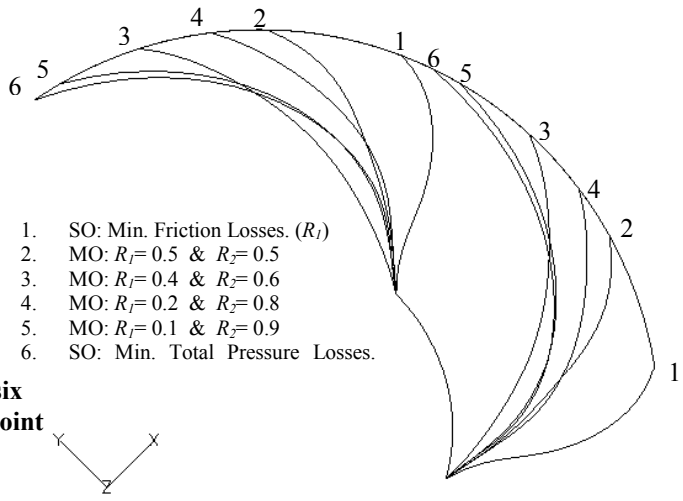

Figure 8, Impeller's profile geometry for six multi-objectives design cases at the design point (Multi-Objectives "weighted sum")