Multi-objective shape optimization of Francis runner using metamodel assisted genetic algorithm

D Chirkov¹, A Filatova¹,², S Polokhin²

¹Institute of Computational Technologies SB RAS, Novosibirsk, Russia
²Novosibirsk State University, Novosibirsk, Russia

d.chirkov@g.nsu.ru

Abstract. Nowadays evolutionary optimization techniques become an integral part of the design process of hydraulic turbine runners. For optimization the shape of the runner is parameterized by a set of geometrical parameters. The objective functions of each runner geometry, being the efficiency and cavitation performance, are evaluated through CFD analysis of 3D flow-field. As usual, some kind of genetic algorithm is used for searching the Pareto front, due to its inherent possibility of parallel evaluation of different runner variants within one generation. This approach requires extensive CFD computations of hundreds and thousands of runner variants. In order to speed up the optimization process a metamodel (approximate, surrogate model) can be utilized. The metamodel provides an approximation of true dependency of the objective functions on the design parameters. The metamodel is built based on the known objective functions for some initial set of points randomly taken from the design space. Once built, it can be used to replace time consuming CFD computations during the optimization process. In the present paper a metamodel, based on Gaussian processes, is implemented and integrated into a multi-objective genetic algorithm. The obtained algorithm is applied for shape optimization of the Francis turbine runner. The number of the design parameters varied from 4 to 24. The objective functions are multi-point efficiency and cavitation characteristics. True values of the objective functions are evaluated using Reynolds-averaged Navier-Stokes computations of the flow field in a reduced turbine domain, including wicket gate, runner and draft tube. These values are used to train the initial metamodel. In order to enhance predictive capabilities of the metamodel, it is retrained periodically during the optimization loop. The results of metamodel-assisted optimization runs are compared to that obtained without metamodel. It is shown that application of metamodel significantly reduces the number of actual CFD computations even for high number of design parameters and thus speeds up the overall optimization process.

1. Introduction

Nowadays evolutionary algorithms are widely used in the design process of hydraulic turbine flow passages [1-6]. However, successful application of these algorithms requires computations of the objective functions, such as efficiency, strength and cavitation characteristics for thousands of geometry variants. Usually, computation of these functions for each variant of the flow passage is carried out through 3D CFD analysis of the flow field using Reynolds-averaged Navier Stokes (RANS) equations. Even with modern CFD tools these computations still require hours of CPU time.
Thus performing the optimization of turbine flow passage requires huge computational resources and time.

One of the modern approaches to reduce the computational time needed for optimization, is the use of metamodels (surrogate, approximate models) [7,8,9]. The metamodel (MM) gives an approximation of the real dependency of the objective functions on the design parameters. In order to build the metamodel a training set of flow passage geometry is first computed using extensive CFD analysis. Once built, it can be used to replace time consuming CFD computations during the optimization process. There are number of metamodels suggested in the literature. Artificial neural networks, polynomial response surfaces, support vector regression, Gaussian regression (or Kriging) are among others. Besides this, there are number of ways how to incorporate the metamodel into the evolutionary algorithm and the whole optimization process. A good review of these techniques is given in [10].

In [7] different surrogate models were compared for draft tube optimization. However, the number of free parameters was limited to 5. In [8,9] a hierarchical metamodel assisted evolutionary algorithm was proposed and applied for optimization of Francis runner. A radial basis function network was used to pre-evaluate and select the promising designs, which then undergo the exact CFD-based evaluation.

In the present paper a metamodel, based on Gaussian processes [11], is utilized to speedup the Francis runner optimization process.

2. Runner geometry parameterization

Efficient geometry parameterization is important for successful optimization of flow passage shape. In the present paper runner parameterization is similar to that, used in [5]. The runner shape is determined by the blade surface and meridian projections of hub, shroud, and inlet and outlet edges of the blade. The blade surface is presented in the following form

$$\mathbf{R}_{\text{blade}}(u,v) = \mathbf{r}(u,v) + d(u,v) \cdot \mathbf{n}(u,v), \quad (u,v) \in [0,1],$$

where \( \mathbf{r}(u,v) = \{R(u,v),Z(u,v),\Phi(u,v)\} \) is the median surface of the blade presented in cylindrical coordinate system, \( \mathbf{n} \) is the unit normal vector to median surface, \( d(u,v) \) is the thickness distribution, \( u \) and \( v \) are the natural parameters of the blade.

The angular coordinate function is presented as the

$$\Phi(u,v) = \Phi_0(u,v) + \tilde{\Phi}(u,v),$$

where \( \Phi_0(u,v) \) is the angular coordinate function for the initial blade, \( \tilde{\Phi}(u,v) \) is the angular deviation of the modified blade from the initial one. This \( \tilde{\Phi}(u,v) \) is subject to parameterization. Here \( \tilde{\Phi}(u,v) \) is taken as a polynomial of \( u \) and \( v \). For example, in bicubic case variation of \( \tilde{\Phi}(u,v) \) is carried out through variation of 16 coefficients \( \varphi_{ij} \) of Bernstein’s polynomial:

$$\tilde{\Phi}(u,v) = \sum_{i=0}^{3} \sum_{j=0}^{3} \varphi_{ij} B_i(u) B_j(v).$$

where

$$B_i(w) = \frac{3!}{k!(3-k)!} w^k (1-w)^{3-k}.$$

In case of 3rd degree polynomial in \( u \) and 5th degree polynomial in \( v \) the number of angular parameters increases to 24.

RZ-projection of the runner is defined by 8 free parameters, as described in [5]. In the present paper thickness distribution \( d(u,v) \) is assumed constant, borrowed from the initial geometry.
3. Numerical method

The main hydrodynamic characteristics of the Francis runner, reflecting its quality and appropriateness for the current project are the high level of the whole turbine efficiency in a given range of operating points, the absence of cavitation, and low level of pressure pulsations in the whole flow passage. For Francis turbines these characteristics can be computed using RANS models with sufficient accuracy.

During the optimization true objective functions (being the turbine efficiency) were evaluated through the numerical solution of steady state RANS equations, closed by the standard $k-\varepsilon$ turbulence model. Governing equations were solved numerically using CADRUN solver, based on the implicit finite volume artificial compressibility approach. Third order accurate MUSCL scheme is used for discretization of the inviscid fluxes through cell face, while central differences were used for viscous fluxes. Linearized system of discrete equations is solved using LU-SGS iterations.

CFD computations were performed for a unit turbine (runner diameter $D_1=1m$, head $H=1m$) operated at unit speed $n_1$. Computational domain consists of one guide vane channel and one runner channel (periodic stage approach), figure 1. Block structured mesh consists of 42750 cells for the guide vane channel and 64800 cells for the runner channel, see figure 1. Mixing plane boundary condition is applied on “guide vane – runner” interface with circumferential averaging of flow variables ($p, C_r, C_u, C_z, k, \varepsilon$), where $C_r, C_u, C_z$ are the radial, circumferential and axial components of the velocity vector, respectively.

![Figure 1. Computational domain and mesh. 1 – guide vane, 2 – runner blade.](image)

Boundary conditions are the constant flow angle at the guide vane inlet, radial equilibrium condition for pressure at runner outlet, and a fixed total turbine head $H$. Since spiral casing, stay vanes and draft tube are not included in the computational domain, energy losses in these elements should be evaluated using some empirical formulas. For example, draft tube losses were evaluated as the sum of friction and circulation losses

$$h_{DT} = \xi_0 \frac{(\bar{v}_a)^2}{2g} + \frac{(\bar{v}_c)^2}{2g},$$

where $\bar{v}_m = \frac{Q}{S_{DT,in}}$ is the mean axial velocity, $\bar{v}_c = \frac{1}{Q} \int c_u (v \cdot dS)$ is the mean circumferential velocity in the draft tube inlet, $Q$ is the discharge, $S_{DT,in}$ is the draft tube inlet cross-section. Empirical friction coefficient is taken $\xi_0 = 0.15$.

With this statement of boundary conditions discharge $Q$ is not fixed a-priory and found in the process of CFD solution. Whole turbine efficiency was evaluated using the formula
where $M$ is the runner torque, $\omega$ is the angular velocity of the runner, $H$ is the total turbine head, $\eta_{\text{mech}}$ is the mechanical efficiency, taking into account disk friction losses, while $\eta_{\text{vol}}$ is the volumetric efficiency, taking into account the leakage of the flow by-pass the runner.

4. Optimization algorithm

In the present paper a Gaussian metamodel is integrated into existing multi-objective genetic algorithm (MOGA). Basic MOGA is similar to that proposed in [1].

Current approach for utilization of the metamodel is similar to [12]. The idea is the following. In the beginning $n_0$ points (runner geometry variants) are randomly selected from the design space $D \subset \mathbb{R}^n$, where $n$ is the number of free parameters. Then CFD computations are carried out for each point to obtain exact values of the objective functions. Based on this training sample a Gaussian regression is built for each of the objective functions. For that the functionality of sklearn library on Python is used. In the present investigation a linear combination of Rational Quadratic and Matern covariance functions is used as a kernel [11].

Although initial metamodel is very rough, it is used for implementing the “light” optimization using the basic MOGA. This optimization is very fast, since all the objective functions are computed on the fly using the values, predicted by the metamodel. Also this fact allows to use a rich number of individuals within one generation in order to guarantee the convergence of the approximate Pareto set. From the obtained Pareto front a subset of $n_{\text{recalc}}$ points is selected. For these points the exact values of the objective functions are computed using CFD. Then these selected points are added to the training set and the metamodel is rebuilt (retrained). After that the next “light” optimization is performed using the updated metamodel, and the next $n_{\text{recalc}}$ points are selected. This loop is repeated several times until convergence. Thus the metamodel is being refined each time in the vicinity of current approximation of the true Pareto front. Hereafter this algorithm will be referred as MA-MOGA (Metamodel Assisted Multi-Objective Genetic Algorithm).

5. Optimization results

The developed MA-MOGA was tested for a bi-objective optimization of a Francis turbine runner with specific speed $n_s = 52$. Optimization was carried out for two operating points, OP1 and OP2:

- OP1 is the part load ($Q < Q_{\text{op}}$), guide vane opening $\alpha = 23^\circ$, $n_{11}=70$.
- OP2 is the high load ($Q > Q_{\text{op}}$), guide vane opening $\alpha = 29.7^\circ$, $n_{11}=70$.

The objective functions to be maximized are the whole turbine efficiencies in these OPs, evaluated using (4). The number of free geometrical parameters varied from 4 to 24.

Figure 2 shows the comparison of the computed Pareto fronts for the case of 16 free angular parameters, corresponding to bicubic polynomial (3). RZ-projection of the blade was fixed. For obtaining a converged Pareto front the basic MOGA required computation of 120 generations with 48 individuals in each (total number of costly CFD computations is 5760). Optimization using MA-MOGA was carried out with parameters: $n_0=144$, $n_{\text{recalc}}=72$. Light optimizations that take objective functions only from metamodel had 20 generations with 288 individuals in each. In this case 10 metamodel retrains were enough to obtain the converged Pareto front. Therefore, total number of expensive CFD computations was 144$\times$10$\times$72=864. It can be seen that initial metamodel, built on the first $n_0$ runner geometries, is very rough (figure 2, left). Predicted efficiency values (blue squares) are far from the real efficiencies (red triangles) for the selected points. However, after 10 cycles of retraining the metamodel becomes rather fine, at least in the vicinity of the Pareto front. It is justified by the fact, that the objective functions, predicted by the MM, are very close to their real values, computed using CFD. Moreover, the Pareto front, found using MA-MOGA even excels that of MOGA (the value of efficiency is slightly higher: figure 2, right). It can be explained by the premature
convergence of the basic MOGA to local optimum. Thus, for this case MA-MOGA gave better result using 6 times less number of expensive CFD computations.

Figure 3 shows similar results for the case of 24 free parameters: 16 angular and 8 for RZ-projection. Although some points of the MA-MOGA Pareto front (red triangles) are close to MOGA Pareto (black circles), it is generally lower and sparse. It can be seen that after 10 cycles of retraining the metamodel is still not exact. Predicted efficiency values deviate from the exact efficiency, evaluated using CFD. As a consequence of poor metamodel quality, it was not possible for MA-MOGA to achieve the same level of efficiency as obtained by MOGA. 10 additional cycles of retraining do not improve significantly the result. This premature convergence of MA-MOGA means that metamodel refinement occurs far from the location of true Pareto front in the design space. This is a space for improvement of the current implementation. One solution is to increase the size of the initial training set $n_0$, thus improving predictive capabilities of the initial metamodel. Another way is to alter the interaction of MOGA and metamodel.

Figure 4 shows another example of optimization with 24 free parameters, but corresponding to $(3,5)$ degree polynomial $\Phi(u,v)$. It can be seen that the result is much better than shown in figure 3. The real evaluation of MA-MOGA Pareto front is close to the MOGA Pareto with 5 times reduced the number of costly evaluations. Evidently, the quality of the metamodel and the resulting performance of the MA-MOGA depends heavily on the real dependency of the objective functions on the design parameters, their complexity and smoothness.

![Figure 2](image_url)

**Figure 2.** Bi-objective optimization of Francis runner, 16 free angular parameters. ● is the Pareto front obtained without metamodel (MOGA), □ is the Pareto front resulted in light optimization utilizing a current metamodel (“light” Pareto front), △ are the real values of objective functions for $n_{scale}$ points, selected from the “light” Pareto. Left figure corresponds to the initial metamodel, built on the base of initial training set. Right figure corresponds to the refined metamodel, obtained after 10 cycles of retraining.
6. Discussion and conclusion

A metamodel, based on the Gaussian regression, is integrated into multi-objective genetic algorithm. The resulting algorithm is applied for two-objective efficiency optimization of Francis runner with up to 24 free geometry parameters. It was shown that in case of moderate number of free parameters (less than 16) the developed algorithm requires about 5 times less expensive computations of the objective functions than conventional genetic algorithm. Since all the true functional evaluations, needed to update the metamodel can be performed in parallel, this reduction in number of costly computations means the same reduction of time, needed for the whole optimization process.

Increasing the number of free parameters to 24 generally worsens the efficiency of the developed metamodel-assisted algorithm. However, performance of the algorithm depends on the real dependency of the objective functions on the design parameters. In some cases 5 times reduction of computational cost can also be achieved for 24 free design parameters.
It should be noted that in the present paper efficiency of the turbine was evaluated without CFD simulation of the flow in the draft tube. Instead, draft tube losses were estimated using the empirical formula. It is known that real dependency of the draft tube losses on the velocity profile at the draft tube inlet is much more complex, than that given by that formula. Therefore, it is expected, that in case of efficiency evaluation using accurate CFD computation of the flow-field in the draft tube would increase the complexity of the true objective functions, and thus, negatively affect the performance of the proposed metamodel assisted algorithm. This issue is the focus for ongoing research.

Nomenclature

\( \alpha \) Guide vane opening [degrees]
\( g \) Gravity acceleration [m/s\(^2\)]
\( H \) Net head [m]
\( h_{DT} \) Draft tube energy loss [m.w.c]
\( Q \) Discharge [m\(^3\)/s]

References

[1] Sallaberger M, Fisler M, Michaud M, Eisele K, Casey M 2000 The design of Francis turbine runners by 3D Euler simulations coupled to a breeder genetic algorithm Proc. 20th IAHR Symp. (Charlotte, USA)
[2] Kurosawa S, and Nakamura K 2009 Design Optimization of a High Specific Speed Francis Turbine Using Multi-Objective Genetic Algorithm International Journal of Fluid Machinery and Systems 2(2) pp. 102-109
[3] Flores E, Bornard L, Tomas L, Liu J, and Couston M 2012 Design of large Francis turbine using optimal methods Proc. of 26th IAHR Symposium on Hydraulic Machinery and Systems (Beijing, China).
[4] Enomoto Y, Kurosawa S, and Kawajiri H 2012 Design optimization of a high specific speed Francis turbine runner Proc. of 26th IAHR Symposium on Hydraulic Machinery and Systems (Beijing, China)
[5] Lyutov A E, Chirkov D V, Skorospelov V A, Turuk P A, Cherny S G 2015 Coupled Multipoint Shape Optimization of Runner and Draft Tube of Hydraulic Turbines ASME Journal of Fluids Engineering 137 No. 111302
[6] Chirkov D V, Ankudinova A S, Kryukov A E, Cherny S G, and Skorospelov V A 2018 Multi-objective shape optimization of a hydraulic turbine runner using efficiency, strength and weight criteria Structural and Multidisciplinary Optimization 58(2) pp 627–640
[7] Marjavaara B. 2007 CFD Driven Optimization of Hydraulic Turbine Draft Tubes using Surrogate Models, PhD Thesis, Luleå University of Technology
[8] Georgopoulos H A, Kyriacou S A Giannakoglou K C, Grafenberger P, and Parkinson E 2008 Constrained Multi-Objective Design Optimization of Hydraulic Components Using a Hierarchical Metamodell Assisted Evolutionary Algorithm. Part 1: Theory 24th IAHR Symp. (Foz do Iguaçu, Brazil)
[9] Grafenberger P, Parkinson E, Georgoulou H A, Kyriacou S A, and Giannakoglou K C 2008 Constrained Multi-Objective Design Optimization of Hydraulic Components Using a Hierarchical Metamodell Assisted Evolutionary Algorithm. Part 2: Applications 24th IAHR Symp. (Foz do Iguaçu, Brazil)
[10] Diaz-Manriquez A, Toscano G, Barron-Zambrano J H, and Tello-Leal E 2016 A Review of Surrogate Assisted Multiobjective Evolutionary Algorithms Computational Intelligence and Neuroscience 2016 Article ID 9420460
[11] Rasmussen C E and Williams C K I 2006 Gaussian Processes for Machine Learning The MIT Press
[12] Voutchkov I, and Keane 2006 A multiobjective optimization using surrogates *Proceedings of the International Conference on Adaptive Computing in Design and Manufacture* (The M.C.Escher Company, Bristol, UK) pp 167–175