

Multi-objective Optimization by Using Modified PSO Algorithm for Axial Flow Pump Impeller

H.S. Park and Fu-qing Miao

Abstract Axial flow pumps are one type of blade pump with great flux, lower head, highly fluids flow. This type of pump can be used in agriculture, irrigation and massive water project widely. Impellers are the main and highly sensitive part of the pumps which performs the function by transferring energy to the fluid there by increasing pressure and velocity. In axial flow pump design process, in order to get high performance pump, designers usually try to increase the efficiency (η) and decrease the required net positive suction head (NPSHr) simultaneously. In this paper, multi-objective optimization of axial flow pump based on modified Particle Swarm Optimization (MPSO) is performed. At first, the NPSHr and η in a set of axial flow pump are numerically investigated using commercial software ANSYS with the design variables concerning hub angle β_h , chord angle β_c , cascade solidity of chord σ_c , maximum thickness of blade H. And then, using the Group Method of Data Handling (GMDH) type neural networks in commercial software DTREG, the corresponding polynomial representation for NPSHr and η with respect to design variables are obtained. Finally, multi objective optimization based on modified Particle Swarm Optimization (MPSO) approach is used for Pareto based optimization. The result shows that an optimal solution of the axial flow pump impeller was obtained: NPSHr was decreased by 11.68 % and efficiency was increased by 4.24 % simultaneously. It means by using this method, better performance pump with higher efficiency and lower NPSHr can be got and this optimization is feasible.

Keywords Axial flow pump impeller · Multi-objective optimization · GMDH · Modified PSO algorithm (MPSO)

H.S. Park (✉) · F.-q. Miao
Department of Mechanical and Automotive Engineering, University of Ulsan,
Ulsan, Republic of Korea
e-mail: phosk@ulsan.ac.kr

F.-q. Miao
e-mail: miaofuqing120@gmail.com

1 Introduction

Axial flow pumps are huge consumers of energy in various industries. This kind of blade pumps has characters of great flux, lower head, and high fluids flow. Because of its broad use in agriculture, irrigation and massive water project, so many researchers pay attention to the area of axial flow pump design [1]. So it is essential to improve the efficiency of such equipment through the design optimization. Optimization of axial flow pump is a multi-objective optimization problem rather than a single objective optimization problem that has been considered so far in the literature. Luo et al. [14] investigated an multi-object optimum design for hydraulic turbine guide vane based on NSGA-II algorithm. They tried to minimize the loss of total pressure and maximize the minimal pressure in guide vane. Finally, for the optimized guide vanes, the loss was reduced, and the cavitations performance was improved. Zhang et al. [20] presented a multi-objective shape optimization of helico-axial multiphase pump impeller based on NSGA-II and ANN. They tried to maximum the pressure rise and pump efficiency. After the optimization using NSGA-II multi-objective genetic algorithm, the five stages of optimized compression cells were manufactured and applied in experiment test. The result shows that the pump pressure rise and the pump efficiency have increased which indicated that the method is feasible.

NPSHr and efficiency in axial flow pumps are important objective functions to be optimized simultaneously in a real world complex multi-objective optimization problem. These objective functions are either obtained from experiments or computed using very timely and high cost CFD approaches, which cannot be used in an iterative optimization task unless a simple but effective meta-model is constructed over the response surface from numerical or experimental data [18]. Therefore, modelling and optimization of the parameters is applied in this paper by using GMDH-type neural networks and modified multi-objective Particle Swarm Optimization in order to minimize the NPSHr and maximize the efficiency of the pumps simultaneously.

In this paper, pump NPSHr and efficiency are numerically investigated using commercial software ANSYS. Then based on the numerical results of geometrical parameters and pump two objective functions, GMDH neural networks are applied in commercial software DTREG in order to obtain polynomial models of NPSHr and efficiency. The obtained simple polynomial models are then used in a Pareto based modified multi-objective PSO optimization approach to find the best possible combinations of NPSHr and efficiency, known as Pareto front.

2 Definition of Variables and CFD Simulation of Axial Flow Pump

2.1 Definition of Objective Functions

As present in Sect. 1, both the NPSHr and efficiency in axial flow pump are important objective functions to be optimized simultaneously. The efficiency of a axial flow pump is defined by

$$\eta = \frac{P_e}{P} \quad (1)$$

where P_e is the useful power transferred from pump to the liquid. P_e can be given by

$$P_e = \rho g Q H \quad (2)$$

where P is shaft power.

NPSHr means the required net positive suction head which defines the cavitation characteristic of axial flow pump. It is the energy in the liquid required to overcome the friction losses the suction nozzle to the eye of the impeller without causing vaporization [15]. NPSHr varies with design, size, and the operating conditions [2]. It will lead to reduction or stop of the fluid flow and damage the pump with the increasing of the NPSHr. In the handbook of pump design, the NPSHr can be calculated with the following formula:

$$\text{NPSHr} = \frac{P_{\text{in}} - P_{\text{min}}}{\rho g} + \frac{v_0^2}{2g} \quad (3)$$

where P_{in} is the pressure at inlet, P_{min} is the minimum pressure at the whole blade surface which can be obtained from the post processing of ANSYS Fluent software simulation [6]. ρ is the density of fluid and v_0 is the inlet velocity.

2.2 Definition of Design Variables

The design variables in this paper are hub angle β_h , chord angle β_c , cascade solidity of chord σ_c ($\sigma_c = l/t$), maximum thickness of blade H . There are two sections are defined in the blades, one on hub and another on shroud as shown in Fig. 1.

So there are four design variables namely: β_h , β_c , σ_c and H . The various designs can be generated and evaluated in ANSYS Fluent by changing the geometrical independent parameters as shown in Table 1. Consequently, some meta-models can be optimally constructed using the GMDH type neural networks in commercial software DTREG, which will be further used in multi-objective Pareto based design

Fig. 1 Design variables of impeller blades

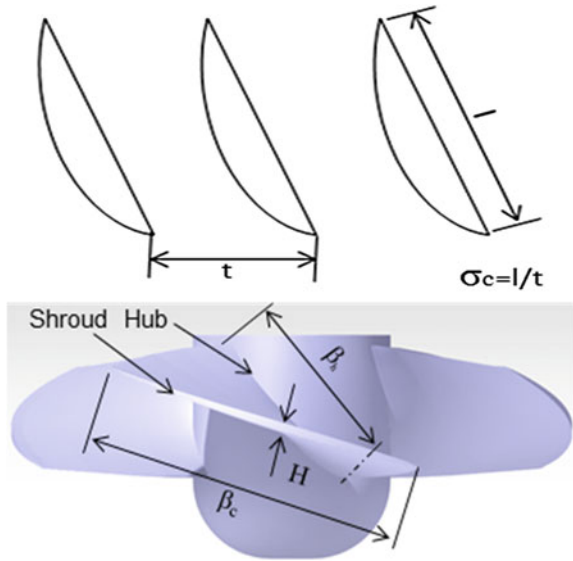


Table 1 Design variables and their range

Design variables	Range from	Range to	Selected values
β_h	36°	54°	36°, 45°, 54°
β_c	21°	25°	21°, 23°, 25°
σ_c	0.75	0.85	0.75, 0.80, 0.85
H	7 mm	11 mm	7 mm, 9 mm, 11 mm

of axial flow pump using modified PSO method. In this way, 81 various CFD analyses have been performed.

2.3 Flow Analysis

Because of the incompressible fluid flow, the equations of continuity and balance of momentum are given as

$$\frac{\partial V_i}{\partial x_i} = 0 \tag{4}$$

$$\frac{DV_i}{Dt} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 V_i}{\partial x_j \partial x_j} - \frac{\partial}{\partial x_j} \overline{u_i u_j} \tag{5}$$

Table 2 Operating conditions in simulation

Parameter	Value
Number of blades	4
Fluid temperature	20°C
Liquid density	998 kg/m ³
Rotation speed	530 rmp
Mass flow rate	2 m ³ /s
Pump head	2.3 m

The physical model that used in the solver is the Reynlds-Averaged Navier-Stokes equations and the k-ε turbulence models is used [17]. The k-ε equations are given as

$$\frac{Dk}{Dt} = \frac{\partial}{\partial x_j} \left[\left(C_k \frac{k^2}{\varepsilon} + \nu \right) \frac{\partial k}{\partial x_i} \right] - \overline{u_i u_j} \frac{\partial V_i}{\partial x_j} \quad (6)$$

$$\frac{D\varepsilon}{Dt} = \frac{\partial}{\partial x_j} \left[\left(C_k \frac{k^2}{\varepsilon} + \nu \right) \frac{\partial \varepsilon}{\partial x_j} \right] - C_{\varepsilon 1} \frac{\varepsilon}{k} \overline{u_i u_j} \frac{\partial V_i}{\partial x_j} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} \quad (7)$$

For the grid generation, tetragonal, hexagonal mesh type was used in software ANSYS Gambit 2.4.6. Pump and around of impeller areas used tetragonal mesh, the other areas were filled with hexagonal mesh [9].

Boundary conditions are as follows: non-slip conditions was applied all around the walls, mass flow rate at the pumps inlet, static boundary condition is used at the outlet. The simulation is continued until the solution converged with a total residual of less than 0.0001 [8].

Operating conditions are shown in the Table 2.

The results of numerical simulation using ANSYS Fluent are shown in Table 3, moreover a path line and total pressure contour of the simulation are shown in Figs. 2 and 3. Total 81 CFD simulation results can be used to build the response surface of both the efficiency and the NPSHr using GMDH type neural networks. Such meta models can be used for the Pareto based multi-objective optimization of the axial flow pump using modified PSO method.

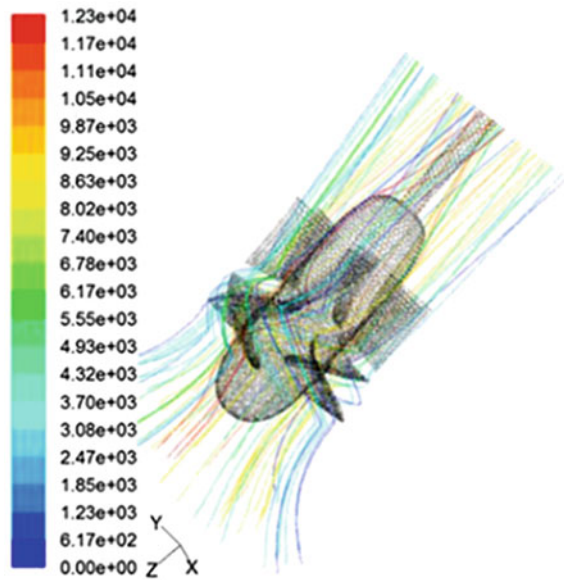
3 Meta-models Building Using GMDH-Type Neural Network

Group Method of Data Handling (GMDH) polynomial neural network are self-organizing approach by which gradually complicated models are generated based on the evaluation of their performances on a set of multi input-single output data

Table 3 Numerical results of CFD simulation

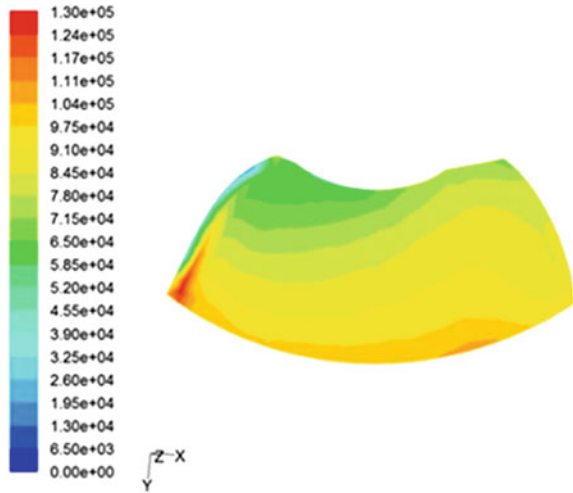
No.	Design Variables				Objective	
	β_h (°)	β_c (°)	σ_c (°)	H (mm)	NPSHr (m)	η (%)
1	36	21	0.75	7	6.5	61.2
2	54	23	0.75	9	7	80.6
3	45	25	0.85	11	7.2	79.4
4	54	21	0.8	11	8.2	78.1
5	36	23	0.75	9	6.2	83.4
6	36	25	0.85	7	7.1	86.4
7	45	21	0.8	9	6.7	80.6
8	45	23	0.85	11	8.4	79.1
9	54	25	0.75	7	8.5	83.4
–	–	–	–	–	–	–
80	54	25	0.85	11	6.5	64.3
81	45	23	0.75	9	7.7	74.1

Fig. 2 Path line of CFD simulation



pairs [12]. GMDH networks were originated in 1968 by Prof Alexey G. Ivakhnenko who was working at that time on a better prediction of fish population in rivers at the Institute of Cybernetics in Kyiv (Ukraine). This algorithm can be used to model complex system without having specific knowledge of the system. The main idea of GMDH is to build an analytical function in a feed forward network based on a

Fig. 3 Total pressure contour of CFD simulation



quadratic node transfer function [5] whose coefficients are obtained using regression technique.

3.1 Structure of a GMDH Network

The meaning of self-organizing means the connections between neurons in the network are not fixed but rather are selected during training to optimize the network. The number of layer in the network also is selected automatically to produce maximum accuracy without over fitting.

As shown in Fig. 4, the first layer (at the left) presents one input for each predictor variable. Every neuron in the second layer draws its inputs from two of the input variables. The neurons in the third layer draw their inputs from two of the neurons in the previous layer and this progress through each layer. The final layer (at the right) draws its two inputs from the previous layer and produces a single value which is the output of the network [16].

The formal definition of identification problem is to find a function f that can be approximately used instead of the actual one f_a . In order to predict output y for a given input vector $X = (x_1, x_2, x_3, \dots, x_n)$ as close as possible to its actual output y_a . For the given M observation of multi input-single output data pairs, there have

$$y_{ai} = f_a(x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}) \quad (i = 1, 2, \dots, M) \tag{8}$$

For any given input vector $X = (x_1, x_2, x_3, \dots, x_n)$, there have

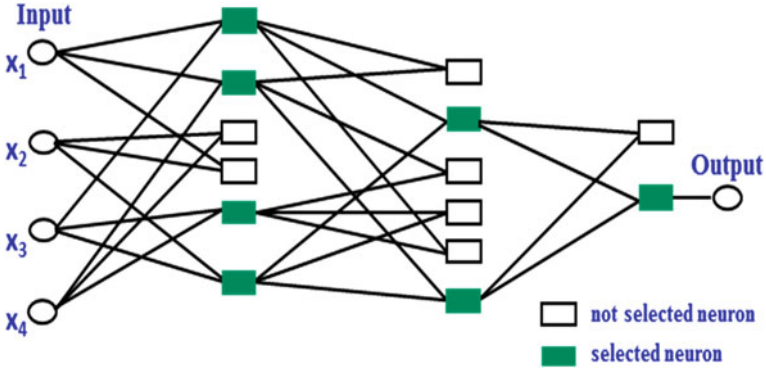


Fig. 4 The structure of a basic GMDH network

$$y_i = f(x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}) \quad (i = 1, 2, \dots, M) \tag{9}$$

In order to determine the GMDH neural network, the square of difference between the actual output and the predicted one is minimized, there have

$$\sum_{i=1}^M [f(x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}) - y_{ai}]^2 \rightarrow \min \tag{10}$$

The most popular base function used in GMDH is the Volterra functional series in the form of

$$y_a = a_0 + \sum_{i=1}^n a_i x_i + \sum_{i=1}^n \sum_{j=1}^n a_{ij} x_i x_j + \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n a_{ijk} x_i x_j x_k + \dots \tag{11}$$

where y_a is the Kolmogorov-Gabor polynomial [5]. It is use complete quadratic polynomials of two variables as transfer functions in the neurons. These polynomials can be represented by the form as show below:

$$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \tag{12}$$

3.2 Meta-models Building in DTREG

The input and output data used in such modelling evolve two different data tables obtained from CFD simulation. Both of the tables consists four variables as inputs that is β_h , β_c , σ_c and H (as shown in Fig. 1) and two outputs that is efficiency η and NPSHr. There have 81 patterns which can be used to train and test GMDH neural network. The corresponding polynomial representation for NPSHr is as follows:

$$N(3) = -0.6432 - 0.015\beta_h + 0.17\beta_c + 0.0325\beta_h^2 - 0.0274\beta_c^2 + 2.3e - 6\beta_h\beta_c$$

$$N(1) = 3.524 - 0.134\beta_h + 0.0245\sigma_c + 0.0418\beta_h^2 + 2.056e - 12\sigma_c^2 + 2.01e - 5\beta_h\sigma_c$$

$$N(7) = 5.643 - 0.2134H - 0.02192 + 0.0021H^2 + 0.0004\beta_h^2 + 2.45e - 5H\beta_h$$

$$N(4) = -1.89 + 0.213\beta_c + 0.015\sigma_c - 0.004\beta_c^2 + 1.19e - 11\sigma_c^2 + 1.24 - 5\beta_c\sigma_c$$

$$N(9) = 6.941 - 3.7845 N(3) + 0.5418N(1) + 0.4723 N(3)^2 + 0.0245 N(1)^2 + 0.0475 N(3) N(1)$$

$$N(6) = 6.147 - 2.1873 N(7) - 0.7122 N(4) + 0.218 N(7)^2 + 0.0947 N(4)^2 + 0.4234 N(7) N(4)$$

$$NPSHr = -0.412 - 0.062N(9) + 1.148N(6) + 0.0812 N(9)^2 + 0.0412 N(6)^2 - 0.1271 N(9) N(6)$$

The corresponding polynomial representation for efficiency is as follows:

$$N(4) = -0.5412 - 0.411\beta_h + 2.143\beta_c + 0.016\beta_h^2 - 0.0124\beta_c^2 + 0.00012\beta_h\beta_c$$

$$N(6) = 16.895 + 1.2183H + 0.495\sigma_c - 0.008H^2 - 0.0041\sigma_c^2 + 0.0012H\sigma_c$$

$$N(1) = -15.03 + 1.96\beta_c + 0.6181\sigma_c - 0.023\beta_c^2 - 0.0041\sigma_c^2 + 0.0012\beta_c\sigma_c$$

$$N(7) = 34.107 + 1.254H - 0.49\beta_h - 0.0082H^2 + 0.0092\beta_h^2 + 0.00059H\beta_h$$

$$N(9) = 54.324 - 0.5912N(4) - 1.0126N(6) + 0.00521N(4)^2 + 0.0063N(6)^2 + 0.0191N(4)N(6)$$

$$N(3) = 61.801 - 0.5012 N(1) - 1.0125N(7) + 0.0043 N(1)^2 + 0.00751N(7)^2 + 0.014N(1)N(7)$$

$$\eta = 0.7241 - 3.0174N(9) + 5.0124N(3) - 2.3104 N(9)^2 - 2.5983N(3)^2 + 5.362N(9)N(3)$$

4 Apply Multi-objective Optimization by Using Modified PSO Method

Particle Swarm Optimization (PSO) is a computational method that optimize a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO is a population-based research algorithm. PSO is originally developed by Kennedy and Eberhart [10]. It was first intended for simulation social behaviour, as a stylized representation of the social behaviour of bird flock or fish school. This algorithm originally adopted for balancing weights in neural networks [4], PSO already became a popular global optimizer. There are one study reported in literature that extend PSO to multi-objective problem [3]. A dynamic neighbourhood particle swarm optimization (DNPSO) for multi objective problems was presented [7]. In their study, for each generation, particles of swarm find their new neighbours. The best local particle in the new neighbourhood is choice as gbest for each particle. A modified DNPSO is introduced by find the nearest n particles as the neighbour of the current particle based on the distances between the current particle from others [19].

PSO algorithm is similar to other algorithms based on the principles are accomplished according to the following equations:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(gb_{ij}^t - x_{ij}^t) \quad (13)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t \quad (14)$$

$$i = 1, 2, \dots, N$$

$$j = 1, 2, \dots, n$$

where x is the particle current position, v is the particle current velocity, t is point of iterations (generations), w is inertia weight, c_1 and c_2 is acceleration constants, r_1 and r_2 is random values range $[0,1]$, pb_{ij} is the personal best position of a given particle and gb_{ij} is the position of the best particle of the entire swarm.

The algorithm developed by Kennedy and Eberhart inspired by the insect swarm (or fish benches or bird flocks) and their coordinated movements. This algorithm pays attention to the information sharing of pb_{ij} and gb_{ij} but just considered the experience of pb_{ij} and gb_{ij} and ignored the communication of other particles. So an improved particle swarm optimization method (IPSO) was developed as the equations shown below:

$$v_{ij}^{t+1} = wv_{ij}^t + c_1r_1(pb_{ij}^t - x_{ij}^t) + c_2r_2(gb_{ij}^t - x_{ij}^t) + c_3r_3CR \quad (15)$$

$$CR = \begin{cases} pb_{kj}^t - x_{ij}^t & \text{if } ran < cp \\ 0 & \text{other} \end{cases} \quad (16)$$

where k means the k th particle and $k \neq i$, cp is communication probability, ran is the random values range $[0,1]$. The communications between particles were considered and could supply much information in order to search optimal solutions in IPSO. Function test indicated that IPSO increase the ability in the search of optimal solutions [13]. But at the later evolution process, with the disappearance of the swarm diversity, the optimization is easier to be trapped into local optimum. For the disadvantages of IPSO, a modified PSO (MPSO) was developed inspired by bacterial foraging algorithm.

Passino originally proposed the Bacterial Foraging Algorithm [11] in 2002. It is inspired by the abstract and simulate of the food engulf of bacterium in human intestinal canal. There have three steps: chemotactic, reproduce and elimination-dispersal to guide the bacterium to the nutrient-rich area.

Elimination-dispersal happened when bacterium got stimulate from outside and then move to the opposite direction. Chemotactic is the action of bacterium gather to the nutrient-rich area. For example, an *E. coli* bacterium can move in two different ways. It can run (swim for a period of time) or it can tumble, and it alternates between two modes of operation its entire lifetime (i.e., it is rare that the flagella will stop rotating).

As introduced, if the flagella rotate clockwise, each flagellum pulls on the cell, and then the net effect is that each flagellum operates relatively independently of the others. Sometimes the bacterium does not have a set direction of movement and there is little displacement.

The bacterium has behaviour of climbing nutrient gradients. The motion patterns that the bacterium will generate in the presence of chemical attractants and repellants are called chemotactic. For *E. coli*, encounters with serine or aspartate result in attractant responses, whereas repellent responses result from the metal ions Ni and Co, changes in pH, amino acids like leucine, and organic acids like acetate [11]. So for the behaviour of draw on advantages and avoid disadvantages, bacterium can search better food source, increase the chance of surviving and enhance the adaptive capacity to varies environment. If there have harmful stimulus in the process of MPSO, it is easy to get rid of local optimal by applying chemotactic operation. The function as shown below:

$$v_{ij}^{t+1} = wv_{ij}^t - c_1r_1(pbest_{ij}^t - x_{ij}^t) - c_2r_2(gbest_{ij}^t - x_{ij}^t) - c_3r_3CR \quad (17)$$

The flow chart of MPSO as shown in Fig. 5:

The polynomial neural network models obtained in Sect. 3 are now employed in a multi-objective optimization procedure using modified PSO method in order to investigate the optimal performance of the axial flow pump. Two conflicting objectives efficiency η and NPSHr that to be simultaneously optimized with the design variables β_h , β_c , σ_c and H .

Design optimization problem of the objective function and constraints as a function of the following equation:

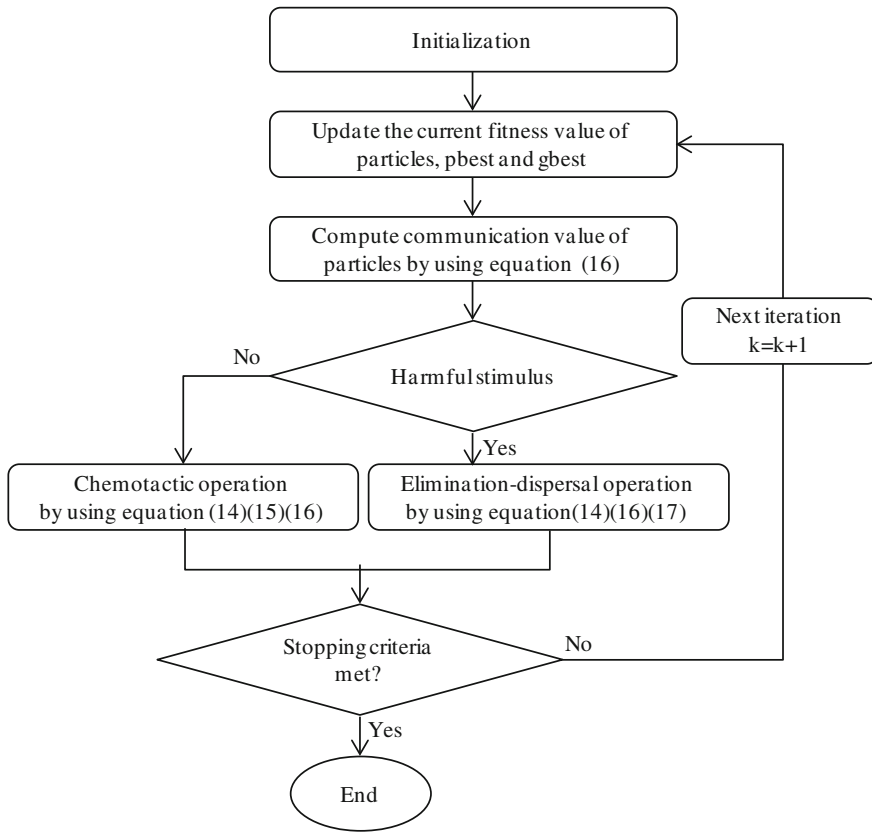


Fig. 5 Flow chart of MPSO

$$\left\{ \begin{array}{l} \text{Maximize} \quad \eta = f_1(\beta_h, \beta_c, \sigma_c, H) \\ \text{Minimize} \quad \text{NPSHr} = f_2(\beta_h, \beta_c, \sigma_c, H) \\ \text{Subject to :} \quad 36^\circ \leq \beta_h \leq 54^\circ \\ \quad \quad \quad 21^\circ \leq \beta_c \leq 25^\circ \\ \quad \quad \quad 0.75 \leq \sigma_c \leq 0.85 \\ \quad \quad \quad 7 \leq H \leq 11 \end{array} \right. \quad (18)$$

The obtained non-dominated optimum solutions based on Pareto front as shown in Fig. 6. These points demonstrate the trade-offs in objective function NPSHr and efficiency. We can find that all the optimum design points in the Pareto front are non-dominated and could be chosen as optimum pump. But choose a better value for any objective function would cause a worse value for another objective. The solutions shown in Fig. 6 are the best possible design points. If we chose any other decision variables, the corresponding values of objectives will be worse.

Fig. 6 Pareto front of NPSHr and efficiency

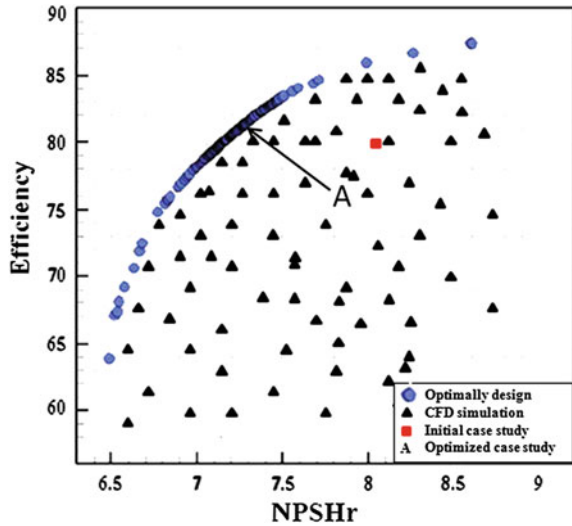


Table 4 Comparison of the results for traditional method and MPSO method

Method	Design variables				Objectives	
	β_h (°)	β_c (°)	σ_c	H (mm)	NPSHr (m)	η (%)
Traditional	48	21	0.75	10.3	8.13	80.2
MPSO	53.7	22.5	0.81	8.2	7.28	83.6

We use the mapping method to find a trade-off optimum design points compromising both of the objective functions.

$$Mapped \text{ value} = \frac{f - f_{min}}{f_{max} - f_{min}} \tag{19}$$

In the mapping method, the value of objective functions of all non-dominated points are mapped into interval 0 and 1. Using sum of these values for each non dominated point, the trade-off point simply is the one having the minimum sum of those values. Consequently, the optimum design point A is the trade-off points which have been obtained from the mapping method. There have a comparison (as shown in Table 4) of the results for traditional method and MPSO method.

In Table 4, comparison of the obtained best compromise solution and the traditional solution as shown. It is clear that in this comparison, NPSHr was decreased by 11.68 % and efficiency was increased by 4.24 % simultaneously.

5 Conclusions

In this paper, multi objective optimization of axial flow pump based on modified Particle Swarm Optimization (PSO) approach are used for Pareto based optimization. Two different polynomial relations for NPSHr and efficiency have been found by GMDH type neural networks using experimentally validated CFD simulations. The obtained polynomial functions were used in an modified PSO optimization process and obtained Pareto front of NPSHr and efficiency. After the mapping method was applied, an optimal solution of the axial flow pump impeller was obtained: NPSHr was decreased by 11.68 % and efficiency was increased by 4.24 % simultaneously. It means this method is feasible and can be applied in impeller design.

References

1. Alision JL, Jiang C et al (1998) Modern tools for water jet pump design and recent advances in the field. International conference on water jet propulsion, RINA, Amsterdam, 1998
2. Bachus L, Custodio A (2003) Know and Understand Centrifugal Pumps. Elsevier press, Amsterdam, p 14
3. Coello CA, Lchuga MS (2002) MOPSO: a proposal for multiple objective particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, Honolulu, Hawaii, USA, pp 1051–1056
4. Eberhart RC, Dobbins R, Simpson PK, (1996) Computational Intelligence PC Tools. Morgan Kaufmann Publishers
5. Farlow SJ (1984) Self-organizing method in modeling: GMDH type algorithm. Marcel Dekker Inc, New York
6. Guan X (1982) Research on development of high performance axial flow pump impellers. Pump Technol 1982(1):25–32
7. Hu X, Eberhart RC, (2002) Multi-objective optimization using dynamic neighborhood particle swarm optimization. In: Proceedings of the IEEE congress on evolutionary computation, Honolulu, Hawaii, USA, pp 1677–1681
8. Huang M, Gao H et al (2008) Numerical simulation and experimental validation of the flow in axial flow pump. Trans Chinese Soc Agric Mach 39(8):66–69
9. Jung IS, Jung WH et al (2012) Shape optimization of impeller blades for a bidirectional axial flow pump using polynomial surrogate model. World Academy of Science, Engineering and Technology, pp 775–781
10. Kennedy J, Eberhart RC (1995) Particle swarms optimization. In: Proceedings IEEE international conference neural networks, vol IV, Perth, Australia, pp 1942–1948
11. Kevin MP (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 22(3):52–67
12. Kristinson K, Dumont G (1992) System identification and control using genetic algorithms, IEEE Trans Syst Manuf Cybern 22(5):1033–1046
13. Lu X, Ding L (2012) An improved particle swarm optimization based on bacterial foraging to airfoil design. Acta Aerodynamica Sinica 30(4):533–538
14. Luo X, Guo P et al (2010) Multi-objective optimum design for hydraulic turbine guide vane based on NSGA-II algorithm. J Drainage Irrig Mach Eng 28(5):369–373
15. Nourbakhsh A (2006) Pump and pumping. University of Tehran Press, Tehran, p 337
16. Phillip HS (2003) DTREG Predictive Modeling Software. DTREG Inc

17. Riasi A, Nourbakhsh A, Raisee M (2009) Unsteady turbulent pipe flow due to water hammer using $k-\epsilon$ turbulence model. *J Hydraul Res* 47(4):429–437
18. Safikhani H, Khalkhali A et al (2011) Pareto based multi-objective optimization of centrifugal pumps using CFD, neural networks and genetic algorithms. *Eng Appl Comput Fluid Mech* 5 (1):37–48
19. Toufik A, Mekki A (2012) Modified PSO algorithm for multi-objective optimization of the cutting parameters. *computer aided engineering, Prod Eng Res Devel* (2012) 6:569–576
20. Zhang J, Zhu H et al (2011) Multi-objective shape optimization of helicon-axial multiphase pump impeller based on NSGA-II and ANN. *Energy Convers Manage*, 538–546