International Journal of
Mathematical Modelling & Computations
Vol. 04, No. 02, 2014, 103- 114

Solving Blasius Equation Using Imperialist Competitive Algorithm

H. Towsyfyan[a,*], S. A. A. Salehi[b], Gh. Davoudi[c], R. Rashidian[d]

[a]*Ph.D. Scholar in Mechanical Engineering,University of Huddersfield,UnitedKingdom.*
[b]*Institute of Standard and Industrial Research of Khorramshahr, Khouzestan, Iran*
[c]*Department of Manufacturing,Technical and Vocational University, Ahvaz, Iran.*
[d]*Department of Computer Engineering,Islamic Azad University, Khoramabad, Iran.*

Abstract. In this study, a new approach is introduced to solve Blasius differential equation using of Imperialist Competitive Algorithm (ICA). This algorithm is inspired by competition mechanism among Imperialists and colonies and has demonstrated excellent capabilities such as simplicity, accuracy, faster convergence and better global optimum achievement in contrast to other evolutionary algorithms. The obtained results have been compared with the exact solution of Blasius equation and another result obtained in previous works and show higher accuracy and less computational requirements. In addition, the method presented with details can be easily extended to solve a wide range of nonlinear problems.

Received: 20 August 2013; Revised: 14 April 2014; Accepted: 7 June 2014.

Keywords: Blasius equation, Imperialist Competitive Algorithm.

Index to information contained in this paper

1. Introduction

Blasius equation is one of the fundamental and basic equations of fluid dynamics, which described the velocity profile of the fluid in the boundary layer theory on a half infinite interval [1, 2]. Many researchers have investigated analytical and numerical solution methods to handle this problem [3, 4]. Most of the work reported on the differential equations is directed to Artificial intelligence. Lee and Kang [12] used parallel processor computers in order to solve a first order differential equation using Hopfield neural

*Corresponding author. Email: towsyfyan@gmail.com

network models. Meade and Fernandez [15] used feed forward neural networks architecture to solve linear and nonlinear ordinary differential equations. Lagaris et al. [11] introduced a new method to solve First order linear ordinary and partital differential equations using artificial neural networks. Malek and ShekariBeidokhti [14] used a hybrid artificial neural network- Nelder-Mead method to solve high order linear differential equations. Lee [13] introduced a new bilaterally approach to find the upper andlower bounds of Blasius equation. A hybrid artificial neural network- swarm intelligence method was used by Khan et al [9] to solve first order nonlinear ODEs. E. Assareh et al. [1] applied Bees Algorithm (BA) method in order to solve Blasius differential equation.

In present study, the basic idea of Imperialist Competitive Algorithm (ICA) is introduced and applied to solve Blasius differential equation which satisfies the boundary conditions. Finally, in order todemonstrate the presented method, a comparison is made between present approach and the solutions which are achieved byLee[13] and Wazwaz[22].

2. Basic Idea of Imperialist Competitive Algorithm (ICA)

In 2007, Atashpaz-Gargari and Lucas [2] introduced the basic idea of Imperialist Competitive Algorithm (ICA) to solve the real world engineering and optimization problems. The proposed algorithm mimics the social–political process of imperialism and imperialistic competition. ICA contains a population of agents or countries. The pseudo-code of the algorithm is as follows.

2.1. Step1: Initial empires creation

Comparable to other evolutionary algorithms, the proposed algorithm starts by an initial population. An array of the problem variables is formed which is called Chromosome in GA and country in this algorithm. In a $N_{var}-$ dimensional optimization problem a country is a $1_\times N_{var}$ array which is defined as follows:

$Country = [P_1, P_2, P_3, \ldots, P_{N_{var}}]$ (1)

A specified number of the most powerful countries, N_{imp}, are chosen as the imperialists and the remaining countries, N_{col}, would be the colonies which are distributed among the imperialists depending on their powers which is calculated using fitness function. The initial empires are demonstrated in Fig.1where more powerful empires have greater number of colonies.

Fig.1. Generating the initial empires: The more colonies an imperialist possess, the bigger is its relevant  mark.

2.2. Step 2: Assimilation policy

To increase their powers, imperialists try to develop their colonies through assimilation policy where countries are forced to move towards them. A schematic description of this process is demonstrated in Fig.2.

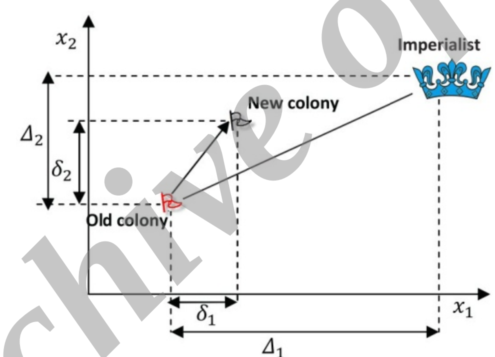

Fig. 2.Movement of colonies toward their relevant imperialist

The colony is drawn by imperialist in the culture and language axes (analogous to any dimension of problem). After applying this policy, the colony will get closer to the imperialist in the mentioned axes (dimensions). In assimilation, each colony moves with a deviation of θ from the connecting line between the colony and its imperialist by x units to increase the search area, where θ and x are random numbers with uniform distribution and β is a number greater than one and d is the distance between the colony and the imperialist state. β>1 causes the colonies to get closer to the imperialist state from both sides.

$$x \sim U(0, \beta \times d) \quad (2)$$

$$\theta \sim U(-\gamma, \gamma) \quad (3)$$

2.3. Step 3: Revolution

In each decade (generation) certain numbers of countries go through a sudden change which is called revolution. This process is similar to mutation process in GA which helps the optimization process escaping local optima traps.

2.4. Step 4: Exchanging the position of imperialist and colony

As the colonies are moving towards the imperialist and revolution happens in some countries, there is a possibility that some of these colonies reach a better position than their respective imperialists. In this case the colony and its relevant imperialist change their positions. The algorithms will be continued using this new country as the imperialist.

2.5. Step 5: Imperialistic competition

The most important process in ICA is imperialistic competition in which all empires try to take over the colonies of other empires. Gradually, weaker empires lose their colonies to the stronger ones. This process is modelled by choosing the weakest colony of the weakest empire and giving it to the appropriate empire which is chosen based on a competition among all empires. Fig.3. demonstrates a schematic of this process.

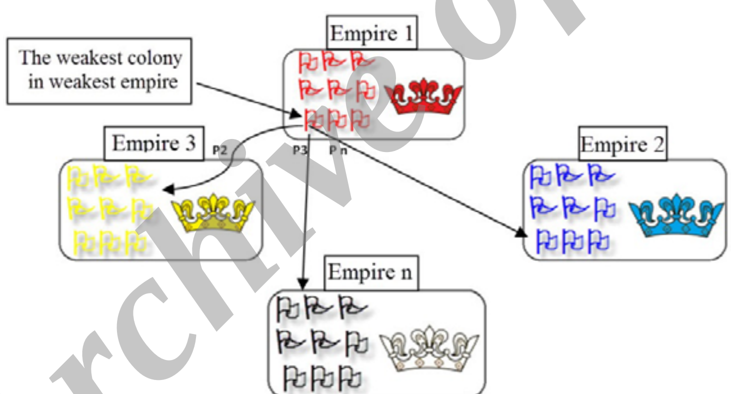

Fig. 3.Imperialistic competition: The more powerful an empire is, the more likely it will possess the weakest colony of the weakest empire.

In this figure empire 1 is considered as the weakest empire, where one of its colonies is under competition process. The empires 2 to *n* are competing for taking its possession. In order to begin the competition, firstly, the possession probability calculated considering the total power of the empire which is the sum of imperialist power and an arbitrary percentage of the mean power of its colonies. Having the possession probability of each empire a mechanism similar to Roulette Wheel is used to give the selected colony to one of the empires considering a proportional probability.

2.6. Step 6: Convergence

Basically the competition can be continued until there would be only one imperialist in the search space, However, different conditions may be selected as termination criteria including reaching a maximum number of iterations or having negligible improvement in objective function. Fig.4. depicts a schematic view of this algorithm. Whenever the convergence criterion is not satisfied, the algorithm continues.

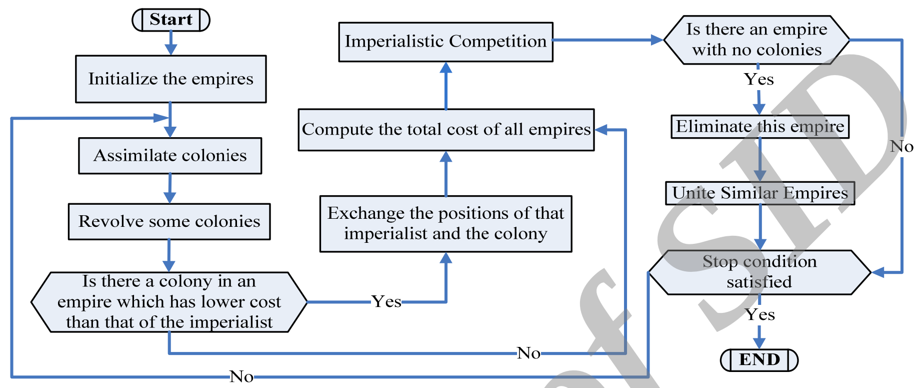

Fig.4. Flowchart of the Imperialist Competitive Algorithm

The main steps of ICA is summarized in the pseudo-code are given in Fig.5. The continuation of the mentioned steps will hopefully cause the countries to converge to the global minimum of the cost function. Different criteria can be used to stop the algorithm.

1-Initialization
    1-1-Set Parameters (PopSize, Number of imperialist,ξ,P-Revolution, % Assimilate)
    1-2-Generating initial Countries (Randomly)
2-Evaluate fitness of each country
3-Form initial empires
    3-1-Choice power countries as imperialists
    3-2-Assigne other countries (colonies) to imperialists based on their power
4-Move the colonies of an empire toward the imperialist (assimilation)
5-Revolution among colonies and imperialist
6- If the cost of colony is lower than own imperialist
    6-1-Exchanging positions of the imperialist and a colony
7- Calculate Total power of the empires.
8-Imperialistic competition
    8-1- Select the weakest colony of the weakest empire and assign this to one of the strange empires
9-Eliminate the powerless empires (the imperialist with no colony)
10-Stop if stopping criteria is met, otherwise go to step 4.

Fig.5. Pseudo code of the Imperialistic Competitive Algorithm

2.7. Literature Review

Imperialist Competitive Algorithm has been successfully applied to solve some engineering problems in recent years, some of those are mentioned below. In Atashpaz-Gargari et al [3] ICA is used to design an optimal controller which not only decentralizes but also optimally controls an industrial Multi Input Multi Output (MIMO) distillation column process. Biabangard-Oskouyi et al. [4] used ICA for reverse analysis of an artificial neural network in order to characterize the properties of materials from sharp indentation test. Nazari et al. [17] solved the integrated product mix-outsourcing (which is a major problem in manufacturing enterprise) using ICA. Kaveh and Talatahari [8] utilized the ICA to optimize design of skeletal structures. Yousefi et al. [23] presented the application of Imperialist Competitive Algorithm for optimization of cross-flow plate fin heat exchanger and concluded that ICA comparing to the traditional GA shows considerable improvements in finding the optimum designs in less computational time under the same population size and iterations. Mozafari et al. [16] applied ICA to optimize intermediate epoxy adhesive layer which is bonded between two dissimilar strips of material. They compared the results of ICA with the Finite Element Method (FEM) and Genetic Algorithm; they showed the success of ICA for designing adhesive joints in composite materials. Towsyfyan et al. have done several works to compare the effectiveness of ICA in comparison with other methods [18-21].

3. Problem Statement

It is well known that the Blasius equation is the mother of all boundary layer equations in fluid mechanics. The differential equation governing the free oscillation of the mathematical Blasius equation, when friction is neglected, is written as follow:

$$\dddot{u}(x) + \frac{1}{2}\, u(x)\, \ddot{u}(x) = 0 \qquad \mathrm{D : x} \ \in [0, \infty]$$

$$u(0) = 0\,, \dot{u}(0) = 0, \dot{u}(\infty) = 1 \ (4)$$

In order to solve Eq. (4), assume a discretization of the domain D with m arbitrary points. Now, the problem can be transformed to the following set of equations:

$$\dddot{u}(x_i) + \tfrac{1}{2} u(x_i)\, \ddot{u}(x_i) = 0 \qquad \forall x_i \in D \qquad , i = 1,2, \dots, m (5)$$

Subject to given boundary conditions. With reference to work of E. Assareh et al. [1] the following nonlinear trial function is assumed as approximate solution:

$$u_T = x - a_1 + a_2\, e^{-x} + a_3\, xe^{-x} + a_4 e^{-2x} (6)$$

Where u_T is trial function and $\vec{a}$ (i.e. a_1 to a_4) are adjustable parameters. These parameters should be determined regarding to minimize the following sum of squared errors, subject to given boundary conditions.

$$Error\ (\vec{a}) = \sum_{i=1}^{m} \left(\ddot{u}_T(x_i) + \frac{1}{2} u_T(x_i) \ddot{u}_T(x_i) \right)^2 \qquad (7)$$

According to given boundary conditions in Eq. 4, following equations are obtained between$a_1\ to\ a_4$.

$$a_3 = 2\,a_1 - a_2 - 1 \qquad (8)$$

$$a_4 = a_1 - a_2 \qquad (9)$$

In order to calculate $Error\ (\vec{a})$ function, derivations respect to independent variable x are needed. Using Eqs (8, 9) required terms in Eq. 7 are written as follows:

$$u_T(x_i) = x_i - a_1 + a_2 e^{-x_i} + (2a_1 - a_2 - 1) x_i e^{-x_i} + (a_1 - a_2) e^{-2x_i} \quad (10)$$

$$u_{\ddot{T}}(x_i) = a_2 e^{-x_i} - (4a_1 - 2a_2 - 2) e^{-x_i} + 4(a_1 - a_2) e^{-2x_i} + (2a_1 - a_2 - 1) x_i e^{-x_i} \quad (11)$$

$$u_{\dddot{T}}(x_i) = (6a_1 - 3a_2 - 3) e^{-x_i} - a_2 e^{-x_i} - 8(a_1 - a_2) e^{-2x_i} - (2a_1 - a_2 - 1) x_i e^{-x_i} \quad (12)$$

Now, the Imperialist Competitive algorithm can be applied in order to determine optimal values of $a_1$and a_2 regarding to minimize Error ($\vec{a}$) function. At the end, $a_3$and a_4 will be calculated from Eqs (8, 9).

***4*. Results and Discussion**

In this study, thecollocation point locations at *xi* were used in interval [0, 5]with step size of 0.5. After very careful investigation, ICA parameters were selected based on table1.

ICA Parameters	
Revolution rate	0.3
Number of Countries	100
Number Of Initial Imperialists	5
Number of decades	100
Assimilation Coefficient (β)	0.5
Assimilation Angle Coefficient (γ)	0.5
Zeta ζ	0.02

Table 1; Parameteres used in ICA.

To choose the proper number of countries for the optimization, the algorithm is executed for different number of initial countries and the respected results for the minimum objective (Error($\vec{a}$)) function at different x_i locations can be seen in Fig.6. Due to the stochastic nature of the algorithm, each execution of the algorithm results in a different result, therefore in the entire study the best solution out of 10 executions is presented as the optimization result. According to Fig.6, it can be seen that in the most *xi* locations the variation of the cost (Error($\vec{a}$)) function is remarkable for the number of countries less than 100. Although more increase in the number of initial countries yields in change in the objective function, the changes are not considerable. Therefore, the number of countries for this study is set to 100 for the rest of the paper.

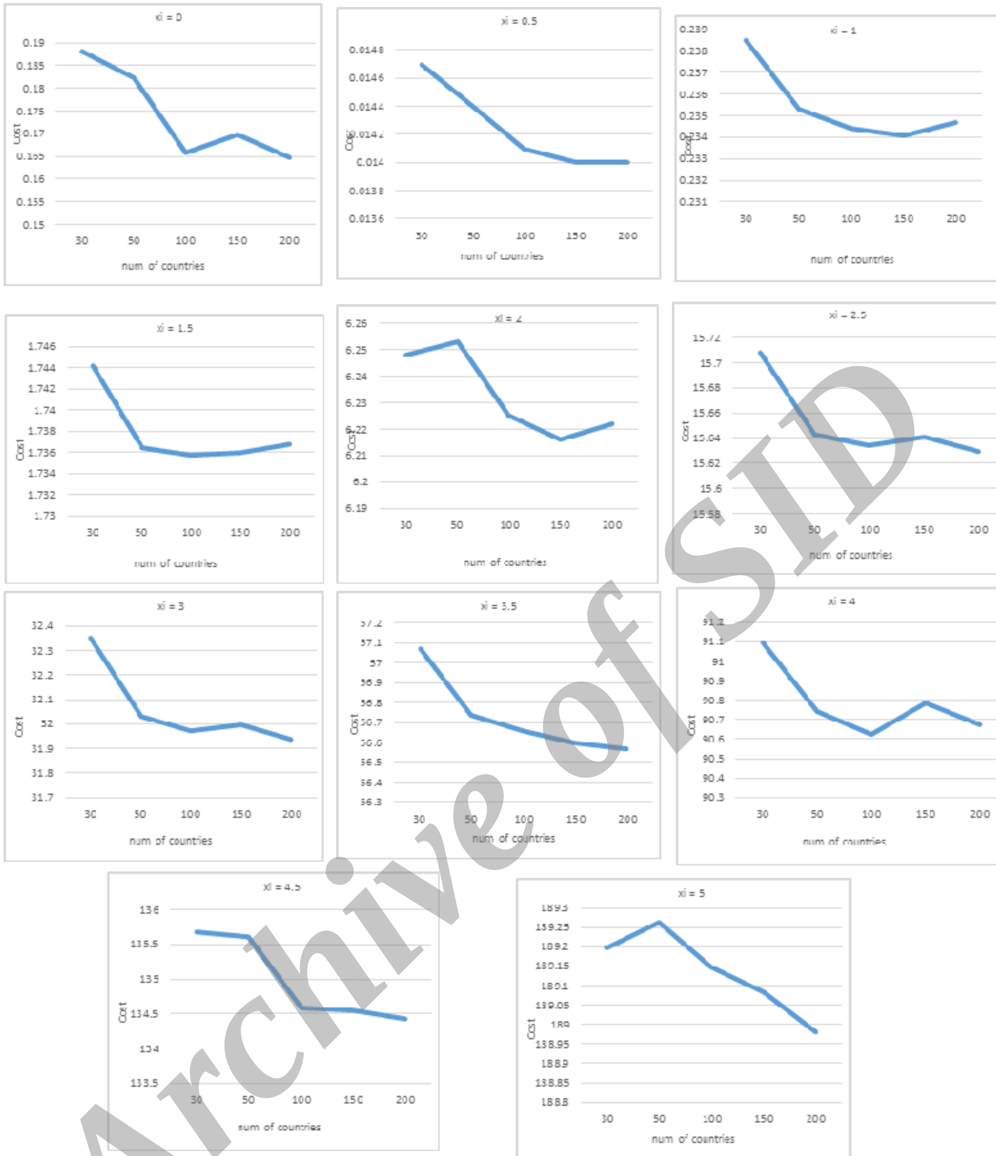

Fig.6. Effect of variation of the number of countries on the Error function at different x_i locations

Fig.7 demonstrates the iteration process of ICA method for optimization of objective (Error($\vec{a}$)) function at different x_i locations in interval [0, 5]. A significant decrease in the target (Error($\vec{a}$)) function is seen in the beginning of the evolution process. After certain decades the changes in the fitness function become relatively minute.

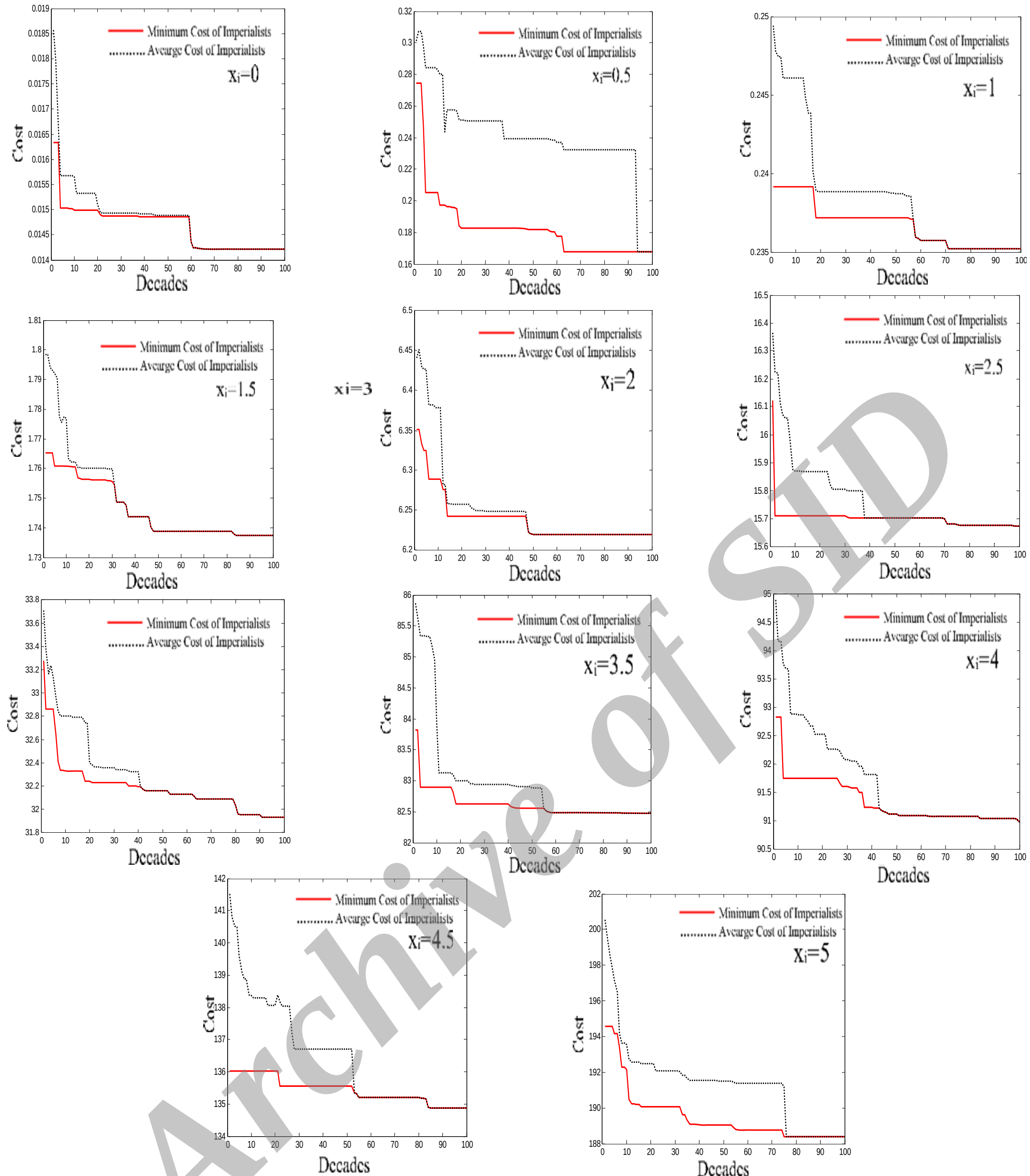

Fig.7. Convergence of minimum the objective function at different locations

In this optimization process, initial number of countries is 100 which 5 of the best ones are chosen to be the imperialists and control others. Since the Optimization problem is finding the optimal values of a_1 and a_2 regarding to minimize $Error\ (\vec{a})$function, the results for the best solution of corresponding adjustable parameters are demonstrated in Table 2.

At the end, a_3 and a_4 will be calculated from Eqs (8, 9).

x_i	$Error(\vec{a})$	a_1	a_2	a_3	a_4	u_T(present method)	Lower bound [9]	Upper bound [9]	VIM [12]
0	0.1672	1.9017	1.5986	1.2048	0.3031	5.55E-17	-0.00053	0.050001	0
0.5	0.014	1.9957	1.5998	1.3916	0.3959	0.1323	0.055298	0.07159	0.0466
1	0.2928	1.9003	1.599	1.2016	0.3013	0.1708	0.161791	0.209158	0.18606
1.5	1.7396	1.9981	1.5978	1.3984	0.4003	0.3464	0.334666	0.449752	0.41563
2	6.2187	1.9021	1.5997	1.2045	0.3024	0.646	0.586389	0.76134	0.72886
2.5	15.7152	1.9061	1.5969	1.2153	0.3092	0.9765	0.904743	1.127819	1.11472
3	31.938	1.9008	1.5982	1.2034	0.3026	1.3593	1.275431	1.535018	1.558191
3.5	56.8064	1.9036	1.5971	1.2101	0.3065	1.7728	1.685283	1.971511	2.04129
4	91.001	1.9008	1.5937	1.2079	0.3071	2.217	2.123457	2.428593	2.54077
4.5	134.9574	1.9011	1.5949	1.2073	0.3062	2.677	2.581608	2.899886	3.01478
5	188.9088	1.901	1.5959	1.2061	0.3051	3.1504	3.053582	3.380849	3.36573

Table 2: the results for the best solution of corresponding adjustable parameters

A careful investigation is carried out to compare the design efficiency of the proposed algorithm with Particle Swarm Optimization (PSO) achieved by Lee [13] and variational iteration method (VIM) achieved by Abdul-Majid Wazwaz [22]. Interested readers may refer to work of Abdul-Majid Wazwaz [22] for a detailed discussion on the VIM method. Fig.8 shows the results of this study, it can be concluded that the results of ICA are in good agreement with previous works.

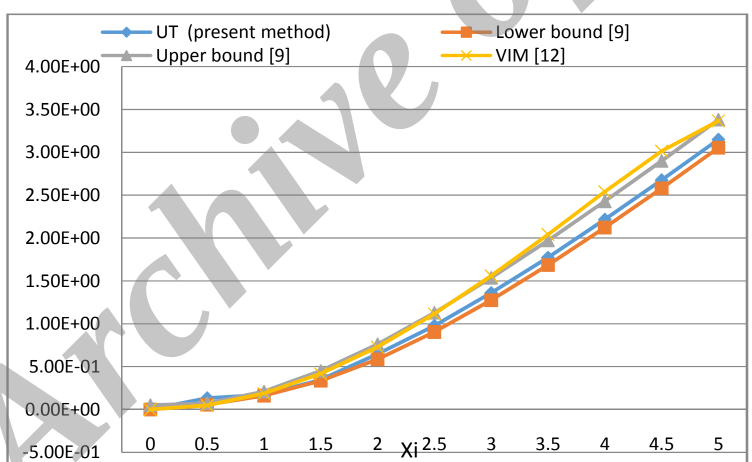

Fig.8. Comparison between obtained results using present method and previous works.

5. Conclusions

In this study, Imperialist Competitive Algorithm (ICA) was applied in order to find the adjustable parameters of approximation function regarding to minimize fitness function. These parameters were determined so that the approximation function has to satisfy the boundary conditions. In order to demonstrate the presented method, a comparison was made between present method here and in the literature. The results showed that our proposed method is in good agreement with obtained solution in the literature. In general, ICA has a promising potential to be used as a new solution approach in a variety of problems.

References:

[1] Assareh, E. Behrang, M.A. Ghalambaz, M. Noghrehabadi, A.R. Ghanbarzadeh, A. A New Approach to Solve Blasius Equation using Parameter Identification of Nonlinear Functions based on the Bees Algorithm (BA). World Academy of Science, Engineering and Technology, **49** (2011) 1109-1111.

[2] Atashpaz-Gargari, E. Lucas, C, Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. IEEE congress on Evolutionary computation, (2007) 4661-4667.

[3] Atashpaz-Gargari, E., Hashemzadeh, F., Rajabioun, R., & Lucas, C. Colonial competitive algorithm, a novel approach for PID controller design in MIMO distillation column process. International Journal of Intelligent Computing and Cybernetics, **1(3)** (2008) 337–355**.**

[4] Biabangard-Oskouyi, A. Atashpaz-Gargari, E. Soltani , N. & Lucas, C., Application of imperialist competitive algorithm for materials property characterization from sharp indentation test. International Journal of Engineering Simulation, 11–12(2009).

[5] Belhachmi Z, Bright B, Taous K. On the concave solutions of the Blasius equations. Acta Mat. Univ. Comenianae LXIX; **2** (2000) 199–214.

[6] Datta BK. Analytic solution for the Blasius equation. Indian J. Pure Appl. Math; **34(2**) (2003) 237–240.

[7] He JH. Approximate analytical solution of Blasius equation, Commun. Nonlinear Sci. Numer. Simul; **3 (4): 260–263** (1998).

[8] Kaveh, A. Talatahar, S., using imperialist competitive algorithm, Computers & Structures, **88, (21–22)** (2010) 1220-1229.

[9] Khan,J.A. Zahoor, R.M.A. Qureshi, I.M. Swarm intelligence for the problem of non-linear ordinary differential equations and its application to well-known Wessinger's equation. European Journal of scientific research. 34(4) (2009) 514-525.

[10] Kuiken HK. On boundary layers in field mechanics that decay algebraically along stretches of wall that are not vanishing small, IMA J. Appl. Math. 27: **387–405** (1981).

[11] Lagris, I.E. Likas, A. Fotiadis, D.I. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks. 9 (5): **987–1000** (1998).

[12] Lee, H. Kang, I.S. Neural algorithms for solving differential equations, Journal of Computational Physics. **91: 110–131** (1990).

[13] Lee, Z.Y. Method of bilaterally bounded to solution Blasius equation using particle swarm optimization. Applied Mathematics and Computation. **179: 779–786**(2006).

[14] Malek, A. Beidokhti, R.S. Numerical solution for high order differential equations using a hybrid neural network—Optimization method. Applied Mathematics and Computation. **183: 260-271**(2006).

[15] Meade Jr, A.J. Fernandez, A.A. The numerical solution of linear ordinary differential equations by feed forward neural networks, Mathematical and Computer Modeling. **19 (12)** (1994) **1–25**.

[16] Mozafari, Hamid. Abdi, Behzad. Ayob, Amran. Optimization of Adhesive-Bonded Fiber Glass Strip using imperialist competitive algorithm, Procedia Technology,Volume 1, 2012, **Pages 194–198**.

[17] Nazari-Shirkouhi, S. Eivazy, H. Ghodsi, R. Rezaie, K. Atashpaz-Gargari, E. Solving the integrated product mix-outsourcing problem using the imperialist competitive algorithm, Expert Systems with Applications, **37(12)** (2010), 7615-7626.

[18] Towsyfyan, Hossein. AdnaniSalehi, Seyed Adnan. Optimization of bead geometry in submerged Arc Welding process using Imperialist Competitive Algorithm, J. Basic. Appl. Sci. Res., 2(12)12582-12589, 2012.

[19] Towsyfyan, Hossein. AdnaniSalehi, Seyed Adnan. Ghayyem, Majed. Mosaedi, Farid Approach to optimization of cutting condition using of imperialist competitive algorithm. International Journal of Mechatronics, Electrical and Computer Technology. Vol. 2(5), pp**144 - 157** (2012), ISSN: 2305-0543.

[20] Towsyfyan,H. Salehi, S, A, A. Davoudi, Gh. and Bahmanpour, M. A New Approach to Solve Differential Equations Arising in Fluid Mechanics,International Journal of Mathematical Modelling &Computations **03, 02**, 2013, **115- 124**.

[21] Towsyfyan,H.Adnani-Salehi, S. A.Ghayyem, M.Mosaedi, F.The Comparison of Imperialist Competitive Algorithm Applied and Genetic Algorithm for Machining Allocation of Clutch Assembly. IJE TRANSACTIONS C: Aspects **26, 12**, (2013), 1483-1492.

[22] Wazwaz, A. M. The variational iteration method for solving two forms of Blasius equation on a half-infinite domain Applied Mathematics and Computation. **188, 485–491**. (2007).

[23] Yousefi, M. Darus, A.N. Mohammadi, H. Exchangers, International Journal of Heat and Mass Transfer, **55(11–12)** (2012) 3178-3185.