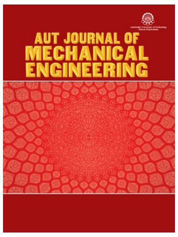


AUT Journal of Mechanical Engineering

AUT J. Mech. Eng., 6(2) (2022) 261-278
DOI: 10.22060/ajme.2021.20376.5998

A New Metaheuristic Method with Applications to Airfoil Shape Optimization

M. Heidari Soreshjani, A. Jahangirian*

Department of Aerospace Engineering, Amirkabir University of Technology, Tehran, Iran

ABSTRACT: This paper proposes an efficient meta-heuristic method called expert groups' optimization algorithm. The method strategy relies on four principles and starts from a random initial population. The population members are divided into two expert groups: the free group and the guided group. Each group has specific tasks for effective domain search, but with one new operator. This operator has an intelligent mechanism so that exploration and exploitation of the population can lead the members to the global optimum. The new method is validated through a standard test function. Then its performance is evaluated in the application of an inverse geometric reconstruction and the results are compared with a genetic algorithm, particle swarm optimization, and mean-variance mapping optimization. Results show that the new method outperforms the alternative methods in convergence rate and reaching the global optimum. Finally, the expert groups' optimization algorithm performance is evaluated in an engineering problem with high computational cost. In this case, the goal is drag coefficient minimization of the RAE 2822 airfoil in transonic flow at a fixed lift coefficient with constraints on the pitching moment and airfoil area. An unstructured grid Navier-Stokes flow solver with a two-equation turbulence model is used to evaluate the aerodynamic objective function. The results show that the optimal solutions obtained by the new method outperform those of mean-variance mapping optimization with considerably faster convergence.

Review History:

Received: Aug. 08, 2021
Revised: Nov. 23, 2021
Accepted: Dec. 21, 2021
Available Online: Dec. 27, 2021

Keywords:

Metaheuristic optimization algorithm
Computationally expensive problem
Aerodynamic shape design
Computational fluid dynamics.

1- Introduction

Many optimization search algorithms have been developed in recent years. Genetic Algorithm (GA) is one of the most powerful methods developed by J. Holland in the 1960s and 1970s. GA evolves a population of candidate solutions for defined objective function using operators inspired by natural genetic variation and natural selection [1]. GA can be used to solve various optimization problems, especially those including issues in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. Simulated Annealing (SA) was developed by S. Kirkpatrick et al. [2] in 1983 inspired by the annealing process of metals. F. Glover [3] was the first to use memory in the meta-heuristic methods in Tabu search. The search history records in a Tabu list in this method and future moves should avoid revisiting previous solutions. In 1992, M. Dorigo et al. [4] proposed a new algorithm called Ant Colony Optimization (ACO). This search technique uses pheromone as a chemical messenger and builds solutions by mimicking the foraging behavior of ants. Such methods are inspired by collective animal behavior known as Swarm Intelligence (SI) methods. Another SI algorithm with significant progress in meta-heuristic algorithms is Particle Swarm Optimization (PSO) by J. Kennedy and R. Eberhart, which gleaned ideas from swarm behavior of bird flocking or fish schooling [5]. PSO has attracted much attention and applied to many optimization problems. One of the critical features of GA and PSO is that they search the design space from a population of members rather than a specific one, resulting in a higher likelihood of finding the globally optimized point. Additionally, they use only the objective function and do not require its derivatives. Such features make GA and PSO attractive for practical engineering applications like aerodynamic shape optimization [6-9]. Although these methods are very successful in finding the global optimum, their major disadvantage is that they require thousands of function evaluations to arrive at solutions with reasonable quality and impose a lot of computational costs, particularly when time-consuming fitness evaluation methods such as Computational Fluid Dynamics (CFD) are used.

At the turn of the 21st century, more diverse meta-heuristic methods were inspired by nature. Harmony search [10], imperialist competitive [11], gravitational search [12], cuckoo search [13], grey wolf optimizer [14], whale optimization algorithm [15], harris hawks optimization [16], atom search optimization [17], and equilibrium optimizer [18] are some of these abundant meta-heuristic algorithms. As can be seen, significant efforts have been made to develop efficient methods for global optimization. However, most of these methods are not specifically developed to optimize computationally

*Corresponding author's email: ajahan@aut.ac.ir expensive functions and perform poorly when faced with limitations on calling the objective function. They usually perform by evolving a population within a relatively large number of fitness evaluations, which could impose an extraordinary computing effort.

Many researchers have proposed surrogate-assisted optimization algorithms to address the problems of expensive optimization. They often use a surrogate method to calculate the objective function with a lower computational cost than the original. These methods are widely used in industry; however, the user's main problem is that achieving sufficient accuracy to estimate the main objective function is often challenging and error-prone. A review of surrogate-based global optimization methods for computationally expensive functions is available in Ref. [19].

There are fewer methods to optimize computationally expensive functions. Tanweer et al. [20] developed an Improved Self-Regulating Particle Swarm Optimization Algorithm (iS-RPSO) for solving computationally expensive numerical optimization problems. In the iSRPSO algorithm, the last two minimum particles use different learning strategies to update speed. These particles are updated from the best particle and the following three particles in search of better solutions. One of the best methods in this field is Mean-Variance Mapping Optimization (MVMO). This method is the best performing algorithm of the expensive optimization competition in CEC2015. MVMO is a novel population-based stochastic optimization algorithm developed by Rueda and István Erlich [21]. The main features of this evolutionary method are in the adoption of a single parent-offspring pair approach along with a normalized search space in the range [0, 1] for all optimization parameters, as well as in the use of a statistical analysis of the evolving solution based on a mapping function. The mapping function is based on the mean and variance of the current best population, and it applies to mutating the offspring. At each iteration, MVMO updates the candidate solution around the best solution. As the search process progresses, the shape and location of the mapping curve are adjusted. Therefore, this method adaptively shifts the search priority from exploration to exploitation. Although the performance of the MVMO is acceptable, it is very complex. Some researchers have proposed a hybrid optimization algorithm. Hybrid Firefly and Particle Swarm Optimization Algorithm is one of these methods that retain the strong points of the combined algorithms [22]. Li et al. [23] developed a Three-Level Radial Basis Function (TLRBF)-assisted optimization algorithm. This method includes three search methods per iteration: global exploration search, subregional search, and local exploitation search. These methods create solutions by optimizing a global Radial Basis Function (RBF) approximation function in the whole search space, in a sub-region, and in the neighborhood of the best current solution, respectively. This method also uses three different algorithms to implement different search methods and has eleven parameters that need to be tuned. These factors cause its users to face special complexities and difficulties. Hence, developing more straightforward and more efficient optimization methods that require a limited number of function evaluations is still an open research issue.

The main contribution of the present work is that, instead of being inspired by nature, some common principles are carefully defined or selected to develop the new method. Then, a fast and efficient optimization procedure is proposed based on these principles. The developed method expects to work well, particularly for optimizing the computationally expensive functions. Details of the new method are presented in the next section, and its performance will be evaluated against three standard test functions, a geometry reconstruction case, and finally an aerodynamic optimization problem.

2- Expert Groups' Optimization Algorithm (EGOA)

2- 1- The optimization principles

Natural phenomena inspire most meta-heuristic methods. However, the main parameter in the succession of a search algorithm is its search principles. In this work, the main principles are defined as follows:

I. All operators should work randomly (or have a random component) to maintain the generality of the method.

II. All population members should have an active and specific role in objective function improvement, and superior members should improve as much as possible in each generation.

III. The optimization method should compromise exploration and exploitation to search all domain regions and gradually focus on the global optimum.

IV. The relationship between members of the population in successive generations should be so that each member can obtain the maximum benefit from the information of prior generations.

Our observations indicate that different meta-heuristic methods vary concerning their commitment to these principles, and more commitments to them cause more success in the optimization method. Therefore, we try to develop an effective method based on the mentioned principles to optimize computationally expensive functions.

2- 2- Algorithm steps

The proposed optimization algorithm is a random population-based method during which the population is split into two expert groups called free search and guided search groups, each includes a specific duty to perform its specified role. Both expert groups use a single comprehensive operator that can manage the balance between the exploration and exploitation of the members. Fig. 1 shows the flowchart of the proposed algorithm.

The proposed method starts from a random population that uses normalized search space within the range [0, 1] for all optimization variables. The initial population forms the basis of the search and, therefore, dramatically affects the search process. The heterogeneous initial population can result in search difficulty or even lack of search in some areas. In addition, if one or more variables are in the order of the truncation error, it can cause problems in the operation of the method. Therefore, using normalized search space in the ini-

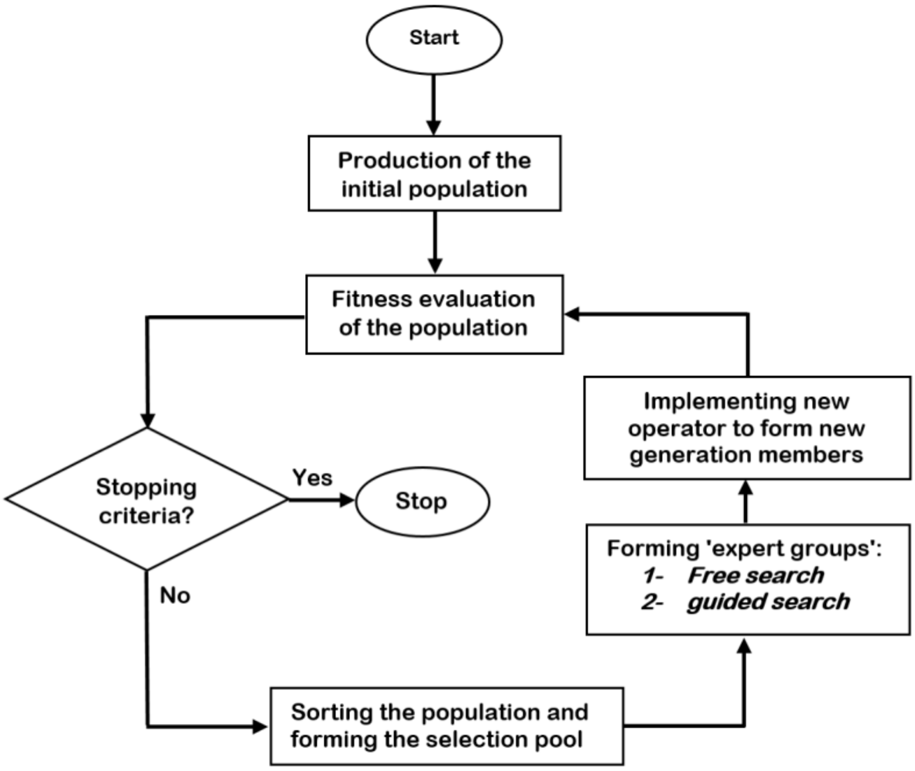

Start
Production of the initial population
Fitness evaluation of the population
Stopping criteria?
Yes
Stop
No
Sorting the population and forming the selection pool
Forming 'expert groups':
1- Free search
2- guided search
Implementing new operator to form new generation members

Fig. 1. Flowchart of the EGOA.

tial population makes the magnitude of the various dimensions in the same order, and all dimensions are searched similarly and without discrimination. After generating the initial population, their objective functions are calculated.

In the second step, members are sorted according to their fitness, and the best member is passed on to the next generation without any changes. Next, a selection pool is made that filters weak population members by only the top half of the members that will be duplicated. Our numerical experiments showed that the presence of weaker population members within the future generations results in slower progress towards the global optimum.

The population is divided into two expert groups within the third step: free search group and guided search group. One of the most important reasons for establishing expert groups is to compromise between exploration and exploitation appropriately. The expert groups can ably manage their members. The free search group is an expert in exploring the domain, while the guided search group is an expert in directing the population to the global optimum regions. An important question here is the population of each group for the efficient implementation of the method. Here a simple strategy follows to allocate more population for the free search group in the early generations. Then they migrate to the guided search group until the end of the method (principle III). A simple trend that showed satisfactory results is the linear function, as illustrated in Fig. 2. The use of groups' variable populations helps EGOA handle the difficulties of balancing between exploration and exploitation.

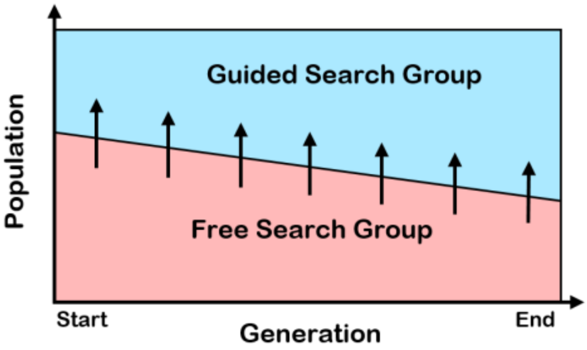


Fig. 2. Population migration from free search group to guided search group.

In the next step, two random permutation members are selected from the selection pool to produce new group members. This random permutation results in the new member parent being non-duplicate and all members of the selection pool selected precisely twice. The 'comprehensive operator' is then applied to the variables of the selected random members to generate the new members. This operator comprises two components: the combination part and the unbalance part (see Eq. (1)). The combination part is the same for both groups, but the unbalance part is specific to each group. The operator selects two random members from the selection pool and combines the variables of these members with a particular proportion. Then a perturbation value is added according to the relation

$$X^{n+1} = c_1 * X_1^n + c_2 * X_2^n + unbalance \tag{1}$$

Where X_1^n and X_2^n are the values of the specific variable corresponding to the first and second member chosen randomly from the selection pool (Principle I) and n is the number of generations, c_1 and c_2 are the ratios of the combination of two member variables, $unbalance$ is a component that is dedicated to each group and explained in the following. This unbalance component should be applied randomly to the combined part of a few variables (for example, 10 to 20 percent of variables).

In this operator, according to a rational rule, it attempts to somewhat move the new members to the best areas found by their parents. The better the value of the objective function, the more prone that area is considered, and the population is directed to that area. Hence, the ratios of the combination of two member variables have a reverse relation with the distance of their fitness to the current best fitness. The ratio of the combination of the first member variables formulated as

$$c_1 = \frac{fitness_2 - best\,fitness}{fitness_1 + fitness_2 - 2 * best\,fitness} \tag{2}$$

Where $fitness_1$ is the first member fitness, $fitness_2$ is the second member fitness, and $best\,fitness$ is the best fitness of all members at the current generation (n). The combination proportion of the second member will be also

$$c_2 = 1 - c_1 \tag{3}$$

In Eqs. (2) and (3), the combination method means that any member closer to the current best member contributes more to creating the new member. This combination method leads to a move towards the superior members of the previous generation and may improve the best members (Principle II). On the other hand, it preserves the effect of weaker members in the generation of new members. Therefore, to some extent, this combination method helps prevent early convergence around the local optimums (Principle III). The unbalance component of the free search group and guided search group can be formulated as

$$free\,unbalance = r * b_f \tag{4}$$

$$guided\,unbalance = r * b_g * \left(X_1^n - best\,X^n\right) \tag{5}$$

Where r is a normally distributed random number (Principle I), b_f is the free group unbalance bound, b_g is the guided group unbalance bound, and $best\,X^n$ is the best member variable values of the current generation. In EGOA, there are only two main internal parameters that must be adjusted: b_f and b_g.

Eq. (4) indicates that the free unbalance term consists of a random number (r) and an unbalance bound (b_f). The random number is the essence of disturbance in unbalance terms and causes randomness (Principle I). The free search via its randomness can help the method not trap the local optimum and resolve early convergence. Another component is the unbalance bound, making it possible for the free group to search freely in a specific bound. Large values of unbalance bound increase the search scope and help the method explore across the field not be trapped in the local optimum, while small values limit the search scope and help the method converge to a particular position. It should be noted that the free search does not mean a random search across the domain, but it spreads the unbalance component around the combined part of the operator. This method preserve the information connection of the population with the previous generation. Hence the members of the new generation continue to benefit from the accumulation experience of previous generations (Principle IV) and avoid wasting a lot of computational costs in irrelevant areas.

Eq. (5) shows that the guided unbalance consists of three components: a random number (r), an unbalance bound (b_g), and a differential component $(X_1^n - best\,X^n)$. The random number and the unbalance bound the same role as in the free unbalance component. However, the magnitude of unbalance bound can be varied in each group. The unbalance bound values of the two groups should be adjusted to compromise between exploration and exploitation of the population (Principle III). We applied a variety of b_f and b_g values and found that the intervals of (0.1 to 0.2) and (0.005 to 0.03), respectively, give the best performance for different test functions. Note that choosing a larger bound practically causes members to spread throughout the field with fewer restrictions, which takes the proposed method away from the goal of using the information of the previous generation (Principle IV). Although this bound may seem limited, the existence of a random initial population, the existence of a random component in all method operators, the allocation of more search members to the free group in the early generations, and the continuation of the free group search until the end of the op-

Table 1. Pseudo-code for the EGOA.

```
begin
    Objective function $f(x)$,   $x = (x_1, ..., x_d)^T$
    Generate an initial population and Evaluate their fitness $Fi$
    while (stop criterion)
        Rank the solutions and keep the best solutions
        Forming the selection pool
        Forming 'Expert Groups':
                1.    Free search group
                2.    Guided search group
        Implementing the operator to form the new generation members
        Evaluate population members fitness $F_i$
    end while
    Post-process results and visualization
end
```

timization process, makes the method to effectively explore all potential areas. The other component that is the main difference between the two groups is the differential component. This component leads to a guided diffusion of members around the combined part of the comprehensive operator. Guided diffusion directs distribution orientation towards the best member of the current generation that moves members towards more susceptible regions (Principle II).

After generating the new members, the objective function is recalculated. If the stopping criteria are satisfied, the optimization process ends, and otherwise, steps 2 to 4 are repeated until the stopping criteria are met. The basic steps of the EGOA can be summarized as the pseudo-code shown in Table 1.

In order to solve a constrained optimization problem with the proposed method, just verify whether the constraints are satisfied for the new member variables. If any constraint is violated, continue generating new members with the proposed operator until a member that satisfies all the constraints is found. If any equality or inequality function exists as a constraint, the penalty function can be added to the objective function. In addition, the weighted sum method, ε-Constraint method, elitist non-dominated sorting algorithm can be used for multi-objective optimization problems.

3- Validation with Standard Test Functions

A numerical experiment is carried out to demonstrate the validation of the expert groups' optimization method. Three well-known and challenging functions named Rastrigin's function, Mikhalewicz's function, and Langermann's function are selected. Rastrigin's function is a separable and highly multimodal function with 10×n local minima on an n-dimensional domain. Due to its large numbers of local minima, finding its global minimum value is a tricky problem. Michalewicz's function is a multimodal test function with n! local optima. The global minimum located at the bottom of valleys, and the values of the function for points outside of narrow peaks, provide very little information about its global optimum. The slope of the valleys is determined by the parameter m . The larger the m , the greater the slope of the valleys and the more difficult it is to find. Whenever m is very large, the function behaves like a needle in a haystack. Langermann's function is also a deceptive function that has many unevenly distributed local minima. According to the reference [24], these functions are defined in Table 2, and their landscape is shown in Fig. 3.

The expert groups' optimization algorithm with a population of 50 and 10 generations is used to find the minimum of this function. Other settings applied to the new method are as follows: $b_f = 0.2$ and $b_g = 0.03$. Fig. 4 illustrates the dispersion of the population within the parameters domains and how they move towards the optimal point. In this figure, the initial population is represented by black circles, the free search group with red asterisks, and guided search group members with blue asterisks. As can be seen, the free search group, which has a large population in the early generations, will explore new areas. At the same time, the guided search group, with a small population, carries out the exploit of previous prime regions. Gradually, the population of the free search group is reduced and is added to the population of the guided search group. Thus, as the process of optimization approaches, the population will be more involved in exploiting identified spaces, and in the end, the entire population is centered on global optimum.

Table 2. Standard test functions.

Function	**Formula**	**Search domain**	**Global minimum**
Rastrigin's function	$f(x)=10n+\sum_{i=1}^{n}\left[x_i^2-10\cos\left(2\pi x_i\right)\right]$	$[-5.12,5.12]$	$f(0,0)=0$
Mikhalewicz's function	$f(x)=-\sum_{i=1}^{d}\sin\left(x_i\right)\left[\sin\left(ix_i^2/\pi\right)\right]^{2m}$; where $m=10$	$[0,\pi]$	$f(2.20,1.57)$ $=-1.8013$
Langermann's function	$f(x)=\sum_{i=1}^{m}c_i\exp\left(\frac{-1}{\pi}\sum_{j=1}^{d}\left(x_j-A_{ij}\right)^2\right)\cos\left(\pi\sum_{j=1}^{d}\left(x_j-A_{ij}\right)^2\right);$ where $m=5$, $c=[1,2,5,2,3]$, $A=[3,5;5,2;2,1;1,4;7,9]$	$[0,10]$	$f(2.79,1.60)$ $=-4.1558$

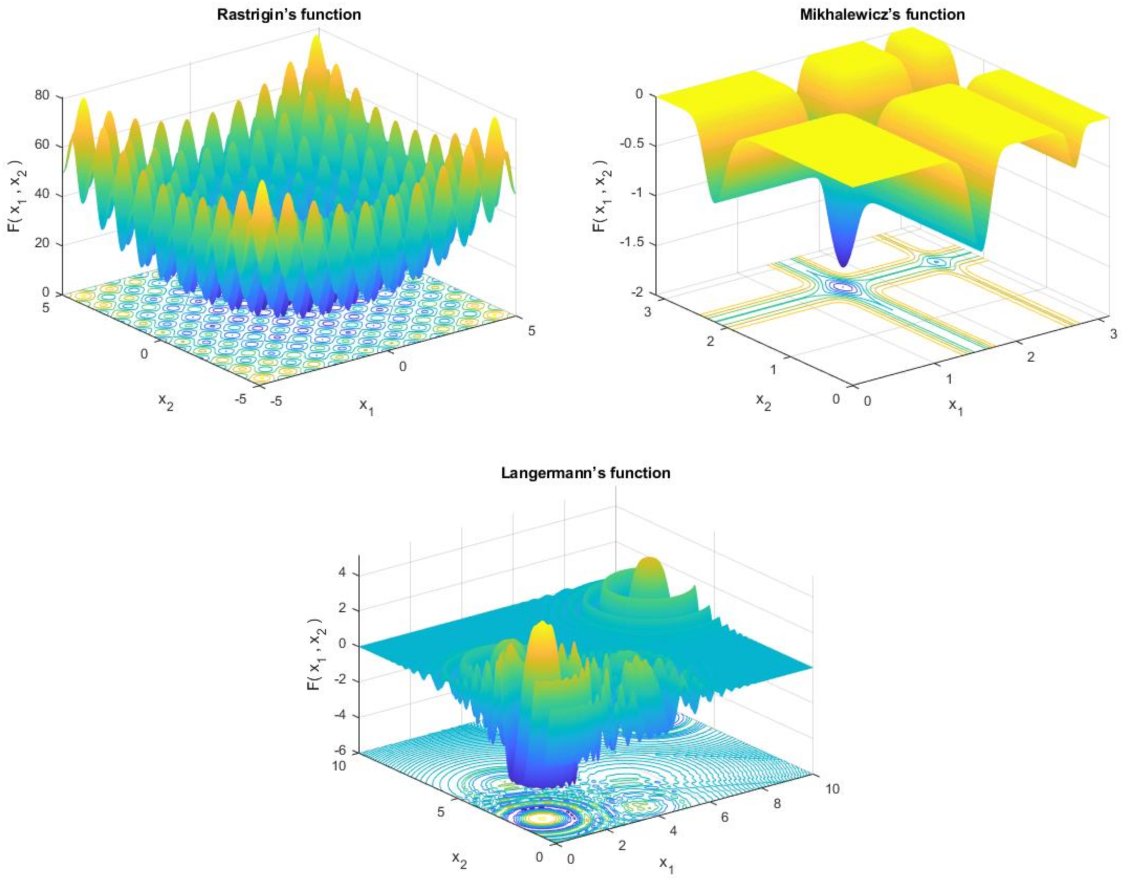
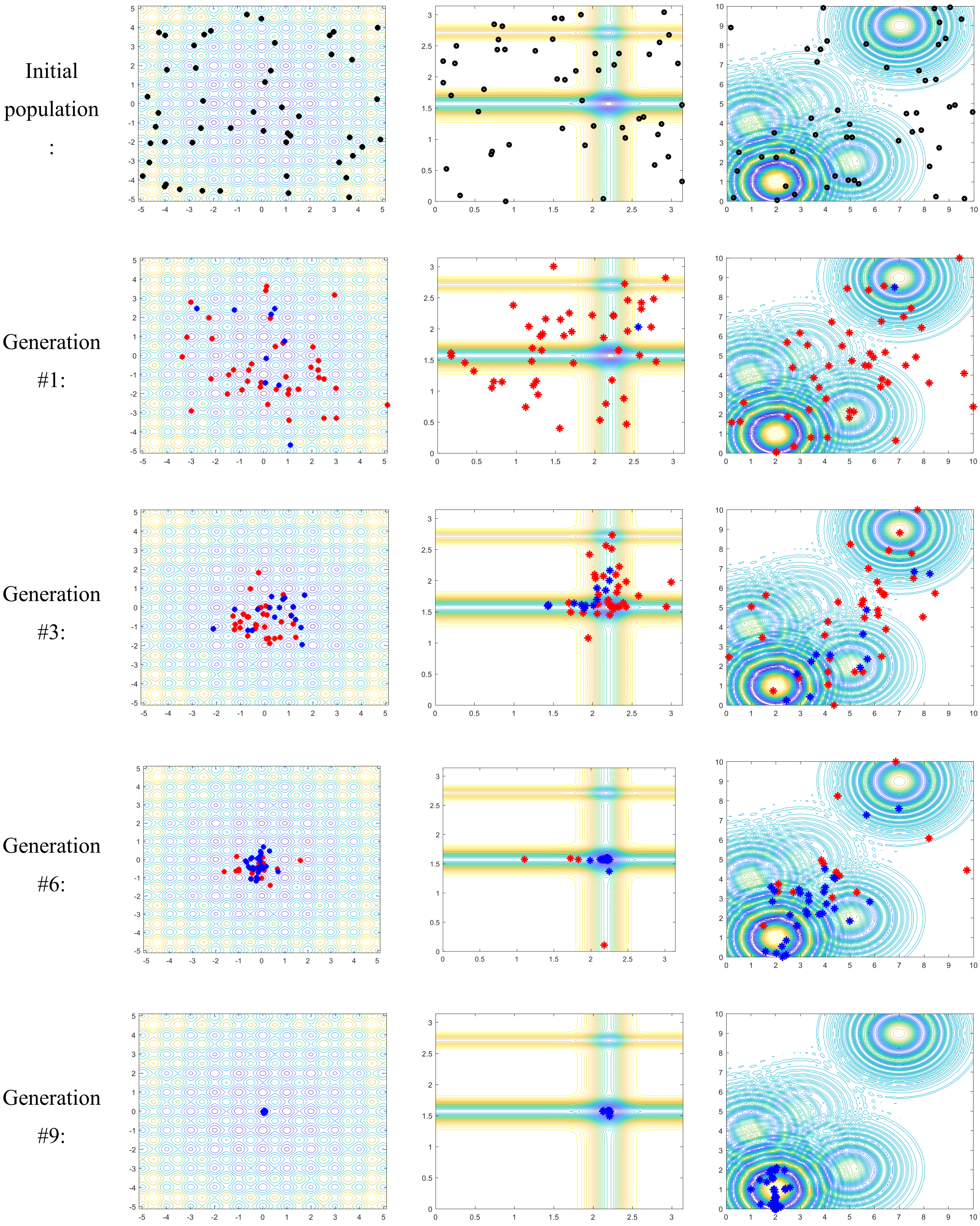


Fig. 3. The landscape of different standard test functions.

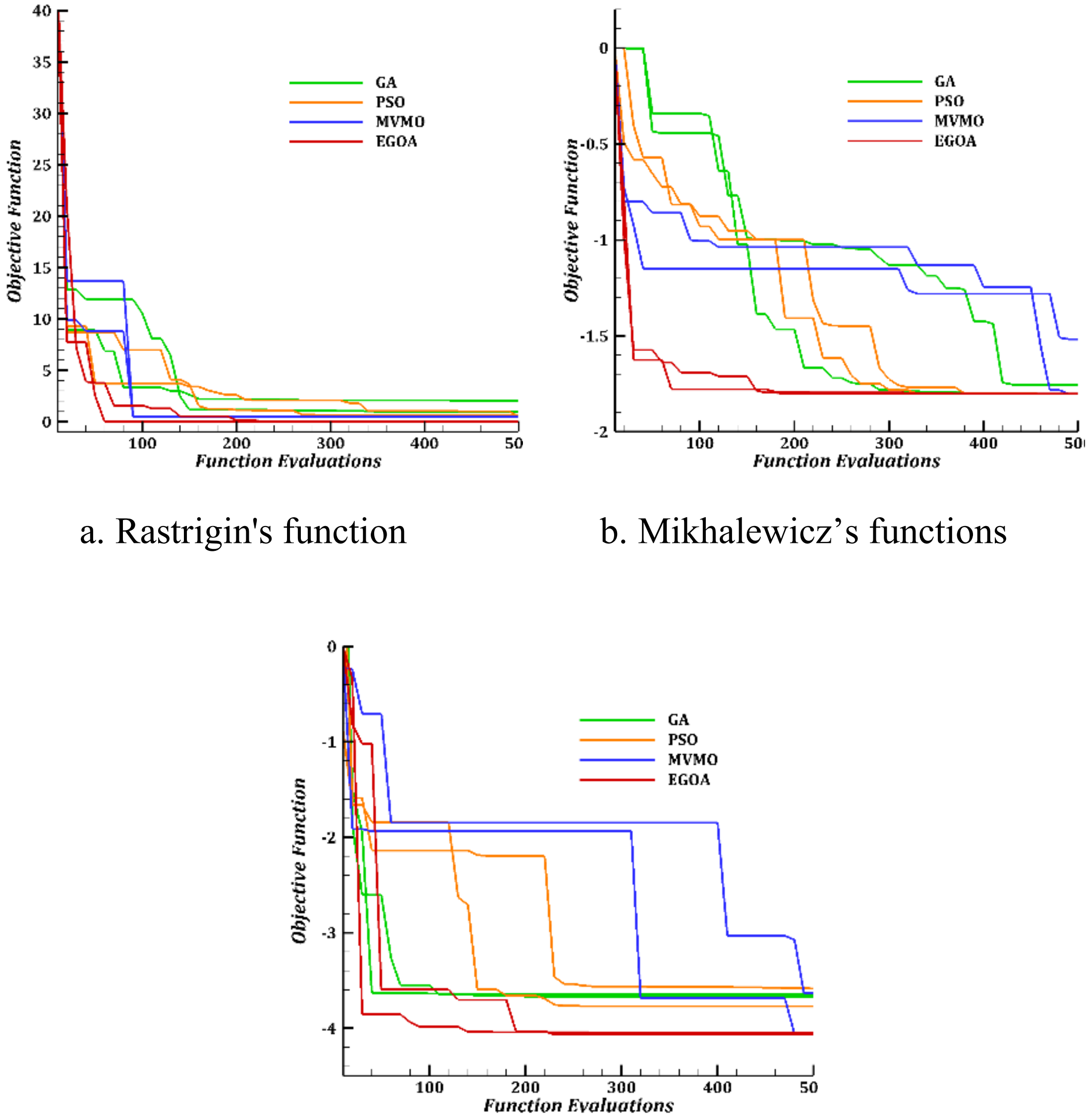


a. Rastrigin's function

b. Mikhalewicz’s function

c. Langermann’s function

Fig. 4. The convergence of members on different functions.



a. Rastrigin's function

b. Mikhalewicz's functions

c. Langermann's functions

Fig. 5. The convergence history of GA, PSO, MVMO, and EGOA.

To better evaluate the new method's performance, results are compared with GA, PSO, and MVMO. GA and PSO methods have been obtained from the MATLAB 2019a optimization toolbox, and the MVMO code has been obtained from [25]. The experiment was repeated twice, and the convergence history of the different methods is shown in Fig. 5. The results show that the proposed method converges more rapidly to the global optimum than other competitors in all three test functions and ultimately achieves better results.

4- Inverse Design Optimization

In this section, the capability and efficiency of the new method are evaluated through the airfoil inverse geometric reconstruction problem, and the results are compared with GA, PSO, and MVMO. It should be noted that only the effects of the optimization algorithms will be studied in this experiment, and all other settings kept the same. The population size of optimization methods is selected as 10. The maximum number of function evaluations is set to 500. All experi-

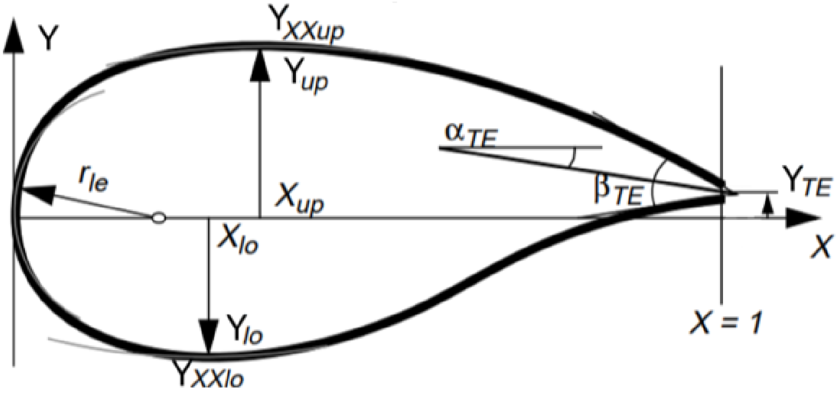


Fig. 6. PARSEC method for airfoil parameterization [26].

Table 3. PARSEC variables and their boundaries.

Variable	*Description*	*Lower boundary*	*Upper boundary*
r_{le}	Leading-edge radius	0.008	0.016
X_{up}	Upper X coordination of crest point	0.30	0.55
Y_{up}	Upper Y coordination of crest point	0.05	0.065
Y_{XXup}	Upper curvature of crest point	-0.48	-0.30
α_{TE}	Trailing edge direction	0	20
β_{TE}	Trailing edge wedge angle	0	16
X_{lo}	Lower X coordination of crest point	0.25	0.40
Y_{lo}	Lower Y coordination of crest point	-0.07	-0.04
Y_{XXlo}	Lower curvature of crest point	0.30	0.90
Y_{TE}	Y coordination of trailing edge	-0.15	0.15

ments are repeated two times to show the general behavior of the methods independent of the random numbers. Other settings are the same as in the previous section. Both methods start from NACA0012 as an initial airfoil, and RAE2822 airfoil is the final target. The objective function is measured as the sum of distances between the obtained and target profiles, as defined in Eq. (6), which should be minimized in the optimization process.

$$O.F. = \frac{\sum_{i=1}^{n_p}\left(Y_i - Y_{ti}\right)^2}{n_p} \tag{6}$$

Where Y_i and Y_{ti} are the design and target coordinates of surface points with fixed X_i coordinate and n_p is the number of surface points.

In this problem, we use the Parametric Section (PARSEC) method to reconstruct the geometry of the airfoil. PARSEC is one of the most common methods that has been used to describe the shape of the airfoils. Fig. 6 shows the ten basic variables of PARSEC, which are the leading edge radius $\left(r_{le}\right)$, upper and lower crest location $\left(X_{up}, Y_{up}, X_{lo}, Y_{lo}\right)$ and curvature $\left(Y_{XXup}, Y_{XXlo}\right)$, trailing edge coordinate, and direction $\left(Y_{TE}, \alpha_{TE}\right)$ and trailing edge wedge angle$\left(\beta_{TE}\right)$. The maximum curvature of the upper and lower surfaces and their location can be effectively controlled using the above variables, which is very useful in reducing the shock wave strength or delaying its occurrence [26]. The PARSEC variables should be bounded to avoid infeasible shapes. These bounds for inverse design are presented in Table 3.

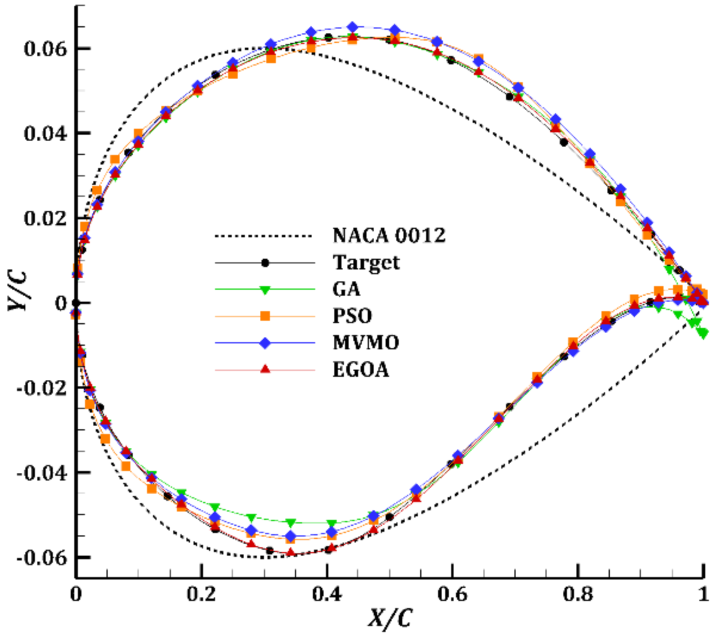


Fig. 7. Target, initial, and constructed geometries after 500 function evaluations.

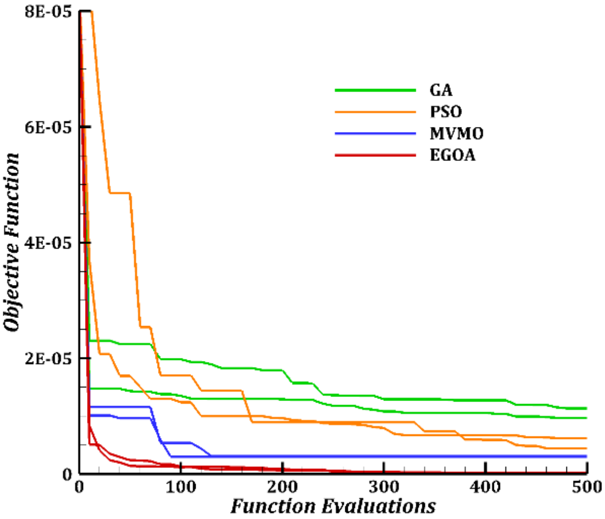


Fig. 8. The convergence history of GA, PSO, MVMO, and EGOA on the inverse design.

Fig. 7 shows the initial airfoil, together with the final airfoils obtained from optimization methods. As can be seen, only the airfoils obtained from EGOA match well with the target airfoil, and other methods have not been able to achieve the desired goal with this limited computational cost. Fig. 8 shows that the convergence rates are different, such that EGOA overcomes other methods in terms of convergence speed and goal achievement. In other words, under the constraint on computational cost, EGOA approaches the target more quickly and accurately, and MVMO ranked second. Fig. 9 shows the new method's intelligent distribution of two variables across their boundaries. This figure also demonstrates how the method manages to scatter members in the early generations and then concentrate around optimal values. As can be seen, the free search will continue continuously until the end of the optimization process. In the next section, the performance of the proposed method is compared with MVMO in an engineering problem with high computational cost.

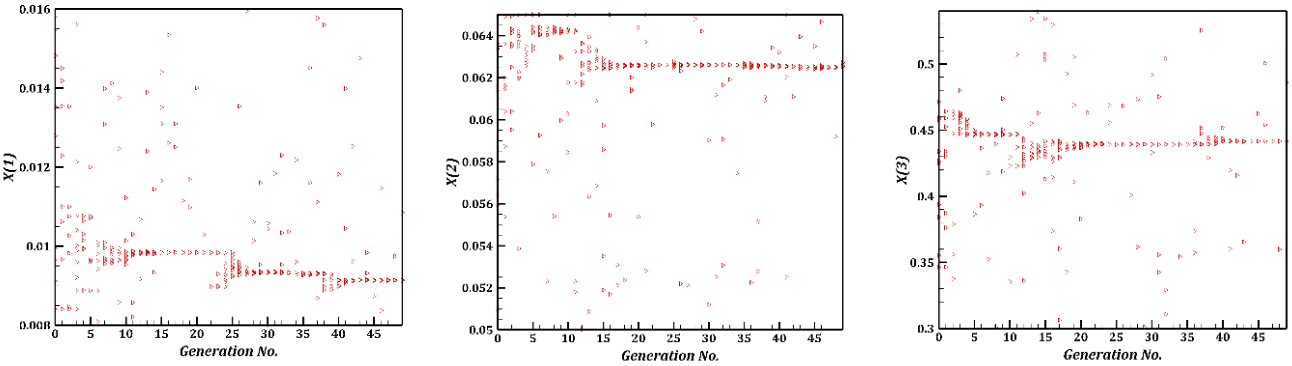


Fig. 9. EGOA scattering history of X (1), X (2), and X (3) after 50 generations.

5- Aerodynamics Optimization

In this section, a numerical implementation is carried out to evaluate the new method's performance in the airfoil shape optimization problems. The problem is the second aerodynamic benchmark case of the ADODG, which is the drag minimization of the RAE 2822 in transonic flow conditions. Because the CFD performs the objective function calculations, the optimization process will be time-consuming and may take days or weeks. Therefore, this test can indicate the capability of the new method in an actual engineering application. The problem is to minimize the drag coefficient of the RAE2822 case 9 at a Mach number of 0.734 and lift coefficient of 0.824 (This approximately corresponds to the wind tunnel corrected the angle of attack of 2.79 degrees), and Reynolds number of 6.5 million concerning area and pitching moment constraints. In this problem, the pitching moment coefficient (evaluated at the quarter-chord) is constrained to $C_m \geq -0.092$. The area must be higher than or equal to the initial RAE2822 airfoil area during the optimization process. In summary, the optimization problem is

$$\textit{Minimize } C_d$$

$$\textit{Subject to}: \begin{cases} C_l = 0.824 \\ C_m \geq -0.092 \\ Area \geq Area_{initial} \end{cases}$$

The lift coefficient constraint is explicitly satisfied by trimming the angle of attack. The pitching moment and the area constraints are handled by using penalty terms in the objective function as defined by

$$\begin{aligned} O.F. = & C_d + abs\left(C_l - 0.824\right) + \\ & 0.5 * \max\left(0, -0.092 - C_m\right) + \\ & \max\left(0, 0.0778 - Area\right) \end{aligned} \tag{7}$$

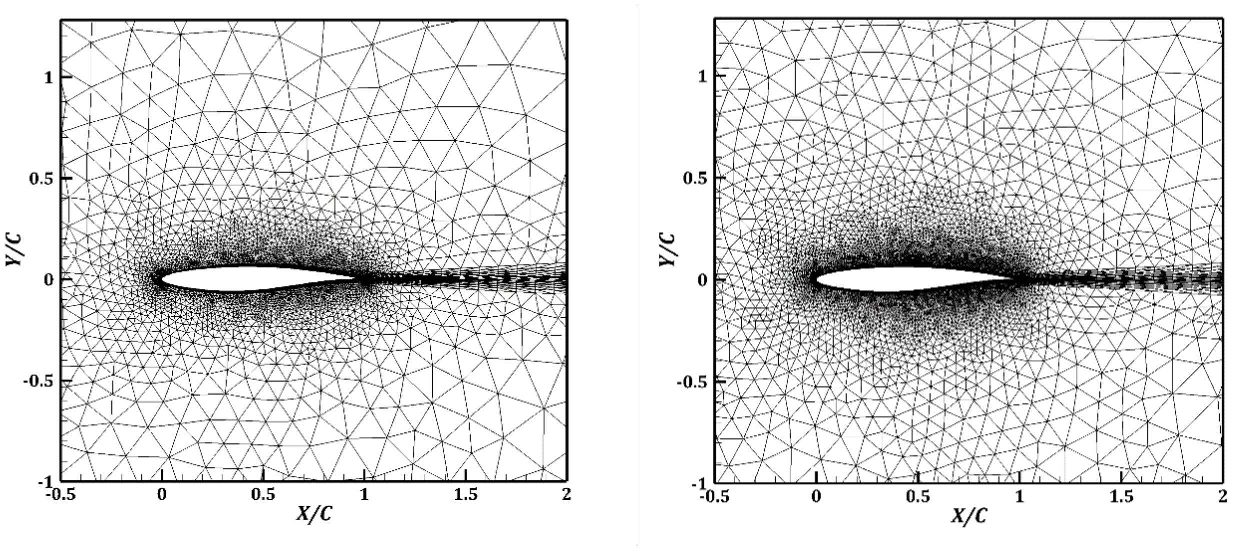
The total population of each generation is set to 10, and the maximum number of generations is 30, so the maximum number of function evaluations is 300. Other settings applied to the new method in the competition are $b_f = 0.1$ and $b_g = 0.01$. The design variables are bounded, as shown in Table 4. As seen in this table, the Y coordinate of the trailing edge is eliminated from the design variables, and the trimmed angle of attack of fluid flow is added instead. In this problem, each experiment is repeated three times, and the best performances of algorithms are compared with each other.

Table 4. Design variables and their boundaries for the aerodynamic benchmark problem.

Variable	*Lower boundary*	*Upper boundary*
r_{le}	0.005	0.012
X_{up}	0.4	0.6
Y_{up}	0.055	0.075
Y_{XXup}	-0.5	-0.3
α_{TE}	2	12
β_{TE}	2	12
X_{lo}	0.3	0.6
Y_{lo}	-0.065	-0.04
Y_{XXlo}	0.4	0.8
Y_{TE}	2.6	3.0

Table 5. Mesh convergence study for initial airfoil (RAE2822)

Grid Level	*No. of Cells*	*C_d (counts)*	*CPU Time (sec)*
Coarse	10651	232.0	567
Medium	18092	230.0	1049
Fine	22299	230.1	1354



a) Coarse grid

b) Medium grid

Fig. 10. Unstructured grids around RAE2822 airfoil.

5- 1- . Grid study

The successive refinement approach proposed by Jahangirian and Johnston [27] generates high-quality viscous grids. The method can produce high-quality (regular) unstructured cells inside the boundary and shear layers and isotropic cells outside these regions. Three grid levels are generated to establish a mesh convergence study. As shown in Table 5, the medium and fine meshes have similar drag coefficients with only a difference in the order 0.1 count. Thus, the medium mesh uses for objective function evaluation since it has lower computational costs than the fine grid. Figs. 10a and 10b show the coarse and medium grids generated around the initial geometry, RAE 2822 airfoil, containing 10651 and 18092 triangular cells, respectively.

In this work, the primary mesh generated around the initial airfoil is moved automatically to fit the newly created airfoils using tension–spring analogy so that there is no need for the user to interrupt the computations . This work provides an automatic and efficient mesh movement tool called hundreds of times during a single optimization process.

5- 2- Objective function evaluation by CFD

The flow solver consumes most of the computational effort required in the airfoil optimization process performed several hundreds of times. Therefore, the CFD solver must possess high efficiency and convergence rate. In this work, the Reynolds Averaged Navier–Stokes flow equations are solved using a finite-volume cell-centered implicit scheme for the unstructured grids [28]. This method reduces the computational time due to convergence acceleration techniques such as residual smoothing and local time steps. Turbulence effects are considered by applying a two-equation k-ε turbulence model with the main flow equations. The wall-function approach is adopted to treat the near-wall region of the boundary layer. The flow solver validation is carried out for the RAE 2822 airfoil using the coarse grid, as this grid is selected for objective function evaluation in this work. The flow conditions are set to the AGARD test case 9 as Mach number of 0.73, Reynolds number of 6.5 million, and incidence angle of 2.79 degrees. Fig. 11 presents the numerical results for the surface pressure coefficient distributions compared to the ex-

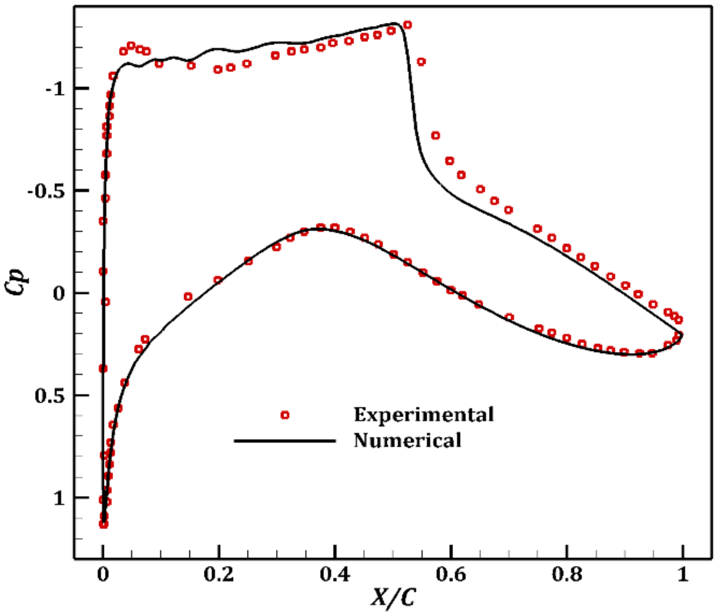


Fig. 11. Surface pressure coefficients for RAE2822 airfoil at M=0.73 and α=2.79 ° [29].

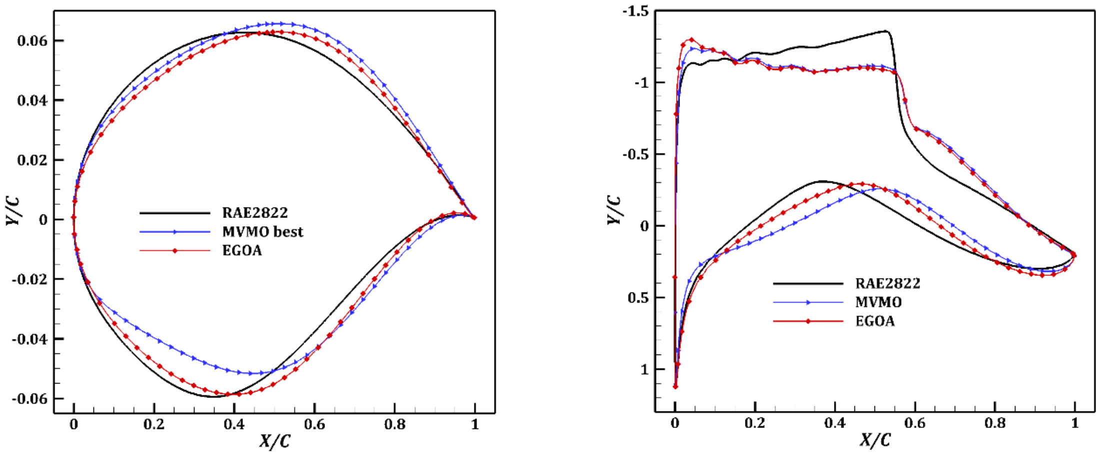
perimental data that shows good agreements [29]. It should be noted that the slight differences between the numerical and experimental data on the upper surface of the airfoil are common in all numerical results.

5- 3- Results

The optimum airfoils at the flow conditions of the second aerodynamic benchmark case of the ADODG are obtained, and their geometry is compared to the initial airfoil in Fig. 12-a. This figure shows that upper surfaces are slightly flattened in both optimum airfoils, which is one of the main features of supercritical airfoils. In addition, their upper crest points are inclined backward and highly downward-curved at the upper trailing edge cause to restore lift loss by flattening the upper surface. However, the lower surfaces of the optimum airfoils are widely different. The airfoil obtained from the new method has a highly curved lower trailing edge, which increases the pressure locally and creates more lift in this area. Instead, in the airfoil obtained from MVMO, the shape of the lower front edge is such as to compensate for the lift deficiency in this area. The surface pressure coefficient distributions are plotted in Fig. 12-b. According to this figure, a strong shock wave appears in the middle part of the initial airfoil upper surface. However, this shock wave is considerably weaker for both optimum airfoils. The weaker shock wave reduces the wave drag and thus reduces the total drag. It should be noted that the two constraints applied to the minimum lift coefficient and the minimum area have caused the shock strength. Fig. 13 shows the Mach number contours around initial and optimum airfoils. As can be seen, there is a sharp shock wave at the RAE2822 upper surface with maximum Mach number 1.341, which is the main flow feature in this case. However, the maximum Mach number is decreased for the present method's and MVMO's optimum airfoils.

The aerodynamics performances of the designed airfoils are compared with those of the initial airfoil in Table 6. Note that the convergence criteria for this case are to reach the same objective function value of about 0.0137. This table indicates that the new method's trimmed angle of attack is 2.789 degrees, giving the target lift coefficient. This angle of attack is 0.16 degrees lower than the angle of attack of the initial airfoil and looks very desirable. In addition, the initial airfoil has a drag coefficient of 0.0230, while the optimum airfoil C_d is only 0.0136. This drag coefficient is in the range of similar work results. However, it should be noted that in this paper, the wall function approach used for near-wall calculations, which generally over-predicts the drag coefficient and using a more accurate turbulence model, can lead to a lower drag coefficient. Besides, the optimum airfoil well satisfied the constraints of the moment coefficient and area. In Table 7, the present method's computational cost and drag coefficient are compared with those obtained by MVMO. As can be seen, MVMO improved the drag coefficient by 40.4% after 2000 function evaluations. However, the proposed method achieved a similar drag coefficient reduction after only 300 function evaluations that take only 15% of the MVMO computational budget.

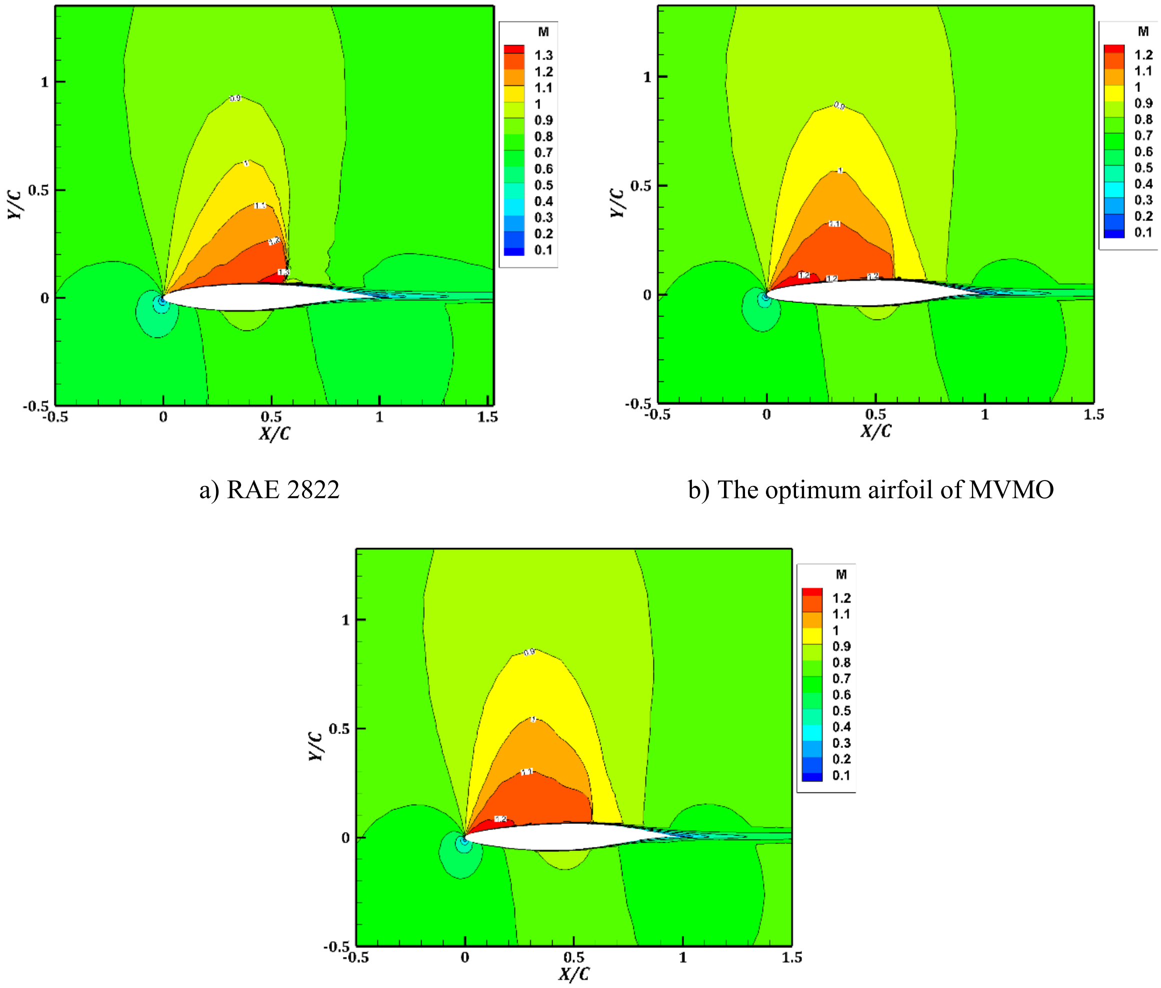
Fig. 14 shows the convergence histories of the objective function and values against the number of function evaluations for three independent implementations of MVMO and the new method. As can be seen, MVMO has a slower convergence with a higher computational cost. However, EGOA has a sharp convergence behavior that shifts the search path toward the optimal point. Hence, the proposed method can achieve similar results at a much lower computational cost due to its higher convergence rate.



a. airfoil shapes

b. pressure distributions

Fig. 12. Comparison of the baseline and optimized airfoils.



a) RAE 2822

b) The optimum airfoil of MVMO

c) The optimum airfoil of the new method

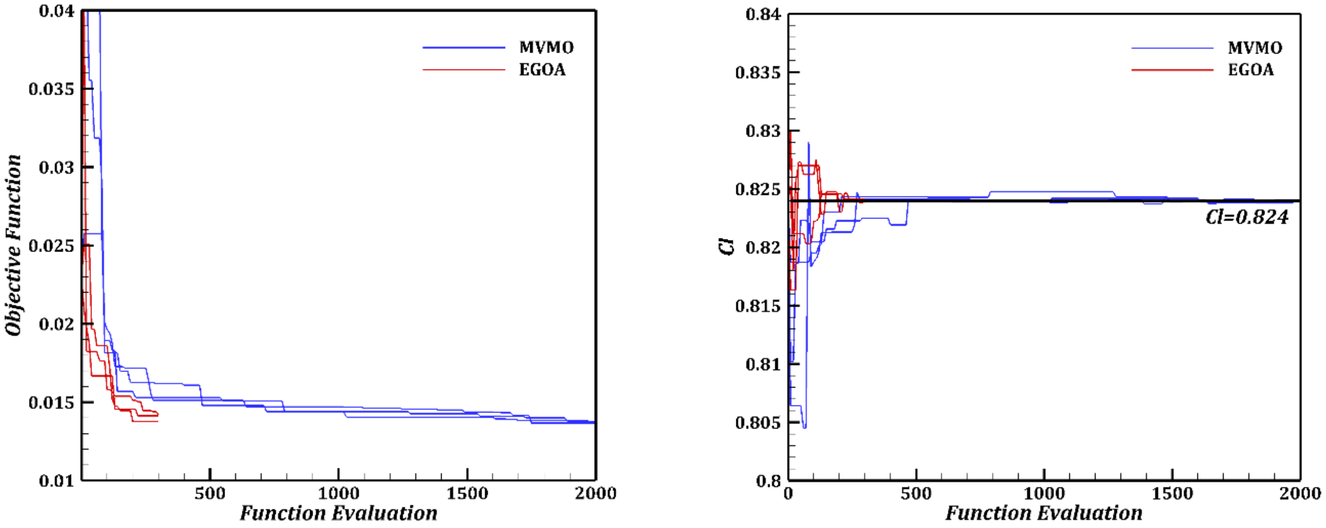
Fig. 13. Comparison of the Mach number contours around initial and optimum airfoils.

Table 6. Results for initial and design airfoils

Parameter	*Initial Airfoil*	*MVMO optimum*	*EGOA optimum*
α_{trim}	2.949	2.798	2.789
C_l	0.8240	0.8240	0.8241
C_d	0.0230	0.0137	0.0136
C_m	-0.0908	-0.0860	-0.0854
Area	0.0778	0.0779	0.0786
Objective Function	0.0230	0.0137	0.0137

Table 7. Comparison of the computational cost and drag coefficient for different methods.

Method	*CFD calls*	*Initial* C_d	*Optimized* C_d	C_d *reduction*	*Computational saving*
MVMO	2000	0.0230	0.0137	40.4%	-
Present Method	300	0.0230	0.0136	40.9%	85%



a. Objective function

b. Cl

Fig. 14. Convergence histories of MVMO against EGOA.

6- Conclusions

A new meta-heuristic optimization method is proposed for computationally expensive problems. It was developed based on four optimization principles. The method was easy to implement and used two expert groups with a simple operator. Using two expert groups made it possible to control and manipulate the population of exploration and exploitation. In addition, the new operator intelligently distributed the population to more susceptible regions, which resulted in higher search population returns. With the help of a comprehensive operator, this property led to the advantage of simplicity and faster convergence. The new method was firstly validated through a standard test function. Then, the method's efficiency was tested compared to GA, PSO, and MVMO for a geometry reconstruction problem. Results showed that the new method outperformed other methods in terms of convergence rate and reaching the global optimum, and MVMO ranked second in this competition. Then, the new method's performance was compared with MVMO for the second aerodynamic benchmark case of the AIAA Design Optimization Discussion Group (ADODG). In this case, the proposed method obtained the same results with 85% lower computational costs compared to MVMO.

References

[1] J. Holland, Adaptation in natural and artificial systems, univ. of mich. press, Ann Arbor, (1975).

[2] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, science, 220(4598) (1983) 671-680.

[3] F. Glover, M. Laguna, Tabu search, in: Handbook of combinatorial optimization, Springer, 1998, pp. 2093-2229.

[4] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE computational intelligence magazine, 1(4) (2006) 28-39.

[5] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95-international conference on neural networks, IEEE, 1995, pp. 1942-1948.

[6] M. Ebrahimi, A. Jahangirian, Accelerating global optimization of aerodynamic shapes using a new surrogate-assisted parallel genetic algorithm, Engineering Optimization, 49(12) (2017) 2079-2094.

[7] D. Kamari, M. Tadjfar, Implementation of Continuous Blowing and Synthetic Jet Actuators to Control the Flow Separation over a Fully Stalled Airfoil, Amirkabir Journal of Mechanical Engineering, 53(9) (2021) 6-6.

[8] M. Kazemi, A. Madadi, M. Mani, Optimization of the Slotted Gurney-Flap geometry applied to NACA 0012 airfoil for aerodynamic performance improvement, Amirkabir Journal of Mechanical Engineering, (2021).

[9] M. Nemati, A. Jahangirian, Robust aerodynamic morphing shape optimization for high-lift missions, Aerospace Science and Technology, 103 (2020) 105897.

[10] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, simulation, 76(2) (2001) 60-68.

[11] E. Atashpaz-Gargari, C. Lucas, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, in: 2007 IEEE congress on evolutionary computation, Ieee, 2007, pp. 4661-4667.

[12] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Information sciences, 179(13) (2009) 2232-2248.

[13] X.-S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World congress on nature & biologically inspired computing (NaBIC), Ieee, 2009, pp. 210-214.

[14] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Advances in engineering software, 69 (2014) 46-61.

[15] S. Mirjalili, A. Lewis, The whale optimization algorithm, Advances in engineering software, 95 (2016) 51-67.

[16] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future generation computer systems, 97 (2019) 849-872.

[17] W. Zhao, L. Wang, Z. Zhang, Atom search optimization and its application to solve a hydrogeologic parameter estimation problem, Knowledge-Based Systems, 163 (2019) 283-304.

[18] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: A novel optimization algorithm, Knowledge-Based Systems, 191 (2020) 105190.

[19] P. Ye, A review on surrogate-based global optimization methods for computationally expensive functions, Software Engineering, 7(4) (2019) 68.

[20] M.R. Tanweer, S. Suresh, N. Sundararajan, Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems, in: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2015, pp. 1943-1949.

[21] J.L. Rueda, I. Erlich, MVMO for bound constrained single-objective computationally expensive numerical optimization, in: 2015 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2015, pp. 1011-1017.

[22] I.B. Aydilek, A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems, Applied Soft Computing, 66 (2018) 232-249.

[23] G. Li, Q. Zhang, Q. Lin, W. Gao, A Three-Level Radial Basis Function Method for Expensive Optimization, IEEE Transactions on Cybernetics, (2021).

[24] M. Molga, C. Smutnicki, Test functions for optimization needs, Test functions for optimization needs, 101 (2005) 48.

[25] P-N-Suganthan, https://github.com/P-N-Suganthan/CEC2015/blob/master/Expensive%20Problems.rar., in, 9 May 2021.

[26] A. Shahrokhi, A. Jahangirian, Airfoil shape parameterization for optimum Navier–Stokes design with genetic algorithm, Aerospace science and technology, 11(6) (2007) 443-450.

[27] A. Jahangirian, L. Johnston, Automatic generation of adaptive unstructured grids for visous flow applications, Mississippi State Univ., Mississippi State, MS (United States), 1996.

[28] A. Jahangirian, M. Hadidoolabi, Unstructured moving

grids for implicit calculation of unsteady compressible viscous flows, International Journal for numerical methods in Fluids, 47(10-11) (2005) 1107-1113.

[29] P. Cook, M. McDonald, M. Firmin, Aerofoil rae 2822-pressure distributions, and boundary layer and wake measurements. experimental data base for computer program assessment, AGARD Report AR, 138 (1979).

HOW TO CITE THIS ARTICLE

M. Heidari Soreshjani, A. Jahangirian, A New Metaheuristic Method with Applications to Airfoil Shape Optimization, AUT J. Mech Eng., 6 (2) (2022) 261-278.

DOI: 10.22060/ajme.2021.20376.5998