AERODYNAMIC SHAPE OPTIMIZATION OF UNGUIDED PROJECTILES USING ANT COLONY OPTIMIZATION AND GENETIC ALGORITHM

Hadi Nobahari*, Seid Y. Nabavi*, Seid H. Pourtakdoust*
** Department of Aerospace Engineering, Sharif University of Technology*

Keywords: *Aerodynamic shape optimization, Genetic Algorithms, Ant Colony Optimization, Continuous Ant Colony System*

Abstract

The problem of aerodynamic shape optimization of unguided projectiles has been investigated. Two stochastic optimization methods have been applied to solve the problem. These include a Genetic Algorithm (GA) and the recently developed Continuous Ant Colony System (CACS), which is based on the well-known Ant Colony Optimization meta-heuristic. The objective function is defined as the summation of normal force coefficients over a set of given flight conditions. An engineering code (EC) is used to calculate the normal force coefficients over the flight conditions. The obtained results of CACS+EC are compared with those of GA+EC, as well as the results of a previous work (GA+AeroDesign). The comparisons show that CACS has superior results compared with GA. It can find better design points in a fewer evaluations. Also there is a good agreement between the results of EC and AeroDesign.

1 Introduction

Some of the key objectives in aerodynamic shape optimization of unguided missiles are to obtain adequate stability in all phases of flight, short minimum range, long maximum range, low gross weight and low dispersion. In practice it is difficult to achieve these objectives due to the complicated dynamic nature of unguided projectiles which are; being nonlinear, time-varying, with some system related uncertainties. There are few analytical studies on external configuration design. The problem is challenging for three reasons. First, the design criteria are themselves functions of aerodynamic and inertial parameters, which in turn are complicated functions of free stream flow conditions, missile geometry and mass distribution. Second, design criteria are often contradictory. Third, design criteria are different for different phases of flight and for different types of unguided projectiles. These difficulties have resulted in the use of numerical optimization methods. In the last two decades, the advent of powerful computers and the resulting advances in soft-computing techniques have encouraged many researchers to use optimization methods as a supplementary design tool.

For many years researchers have applied gradient-based optimization schemes to aerodynamic shape optimization [1-6], but these methods are subject to some limitations. The fact that they must start with an initial set of parameters, can bias future solutions toward a local optimum in the vicinity of the starting point. These methods work most efficiently when there are a small number of design variables and when the variables are essentially independent of each other for these scenarios.

They are able to generate improved designs. However, as the number of variables increases and couplings occur between them, gradient-based algorithms may be trapped into local minima because they do not have the ability to recombine disparate solutions to form solutions that sample a new portion of the search space.

In recent years, there has been an up-growing interest in the use of global optimization methods in a wide range of design problems, as well as the aerodynamic shape optimization. These methods are based on stochastic intelligent operators. That is, rather than starting from a single guess and then marching toward a local optimum based on deterministic transition rules, they test multiple solutions and generate the next solutions using probabilistic transition rules. Since these methods usually do not require gradients or derivatives of the objective function, they can work quite easily on irregular functions such as step and discrete disjointed functions.

Bramlette and Cusic [7] applied Genetic Algorithm (GA) to the parametric design of aircraft. Gage and Kroo [8] applied GA to minimize induced drag of aircraft wings given a fixed lift. They used both a penalty and a repair approach to deal with solutions not achieving the fixed lift value. Anderson [9] has applied GA to subsonic wing design in order to produce good aerodynamic shapes with an additional constraint on the structure. In that work he used penalty weights to combine lift, aerodynamic efficiency, and structural constraints into a global objective function. He pointed out that the achieved solution is strongly dependent on the values of the penalty weights. Later, he removed the weighting procedure and instead used a Pareto GA in the same problem [10]. The results of that work showed that Pareto GA is ideally suited for optimization problems with diverse goals. However, such an approach means that the designer must scan the resulting solutions in a Pareto optimal set to determine which solutions are more desirable. Unlike single objective problems, where there is a clear winner, multi-objective problems require judgment about which solutions are preferable. In [11], a GA was coupled with an inverse design method to design wind turbines that maximize output power at varying wind speeds. In that case, GA executed the design search and the inverse procedure enforced certain constraints while giving the designer flexibility in choosing which variables to iterate with and which to send to GA. Another hybrid approach [12] coupled a GA with a standard gradient approach to maximize lift coefficient for an airfoil. At some prescribed threshold lift coefficient, GA is halted and the gradient method is begun. In Ref [13], GA was used to optimize 15 external ballistic parameters of a missile configuration. In that work, three different objective functions were studied. The first one was defined as the summation of the normal force coefficients at a given flight conditions to be maximized, to see whether GA was able to maximize the fin areas. The AeroDesign software was used to predict the aerodynamic coefficients. The second objective function was defined as the summation of lift to drag forces coefficients and the last one was defined to simultaneously maximize the summation of lift to drag forces coefficients and minimize the summation of static margins. In the last case, Pareto GA was used to tackle the multi-objective optimization problem. In a later work [14], a Parallel Simulated Annealing (PSA) was used to optimize the aerodynamic shape of some internal flow systems, including diffuser shape design, tunnel wall design and axisymmetric nozzle design. A CFD code based on Euler and Navier-Stokes equations was used to compute the flow field.

A problem similar to that of Ref. [13] is considered in this paper. Here the recently developed Continuous Ant Colony System (CACS) [15], which is a variation of the well-known Ant Colony Optimization (ACO) meta-heuristic [16,17] for continuous optimization problems, is used to optimize the external configuration parameters. The aerodynamic coefficients are calculated using an in-house developed Engineering Code (EC). The obtained results are compared with those of [13], as well as the results of GA when it uses EC instead of AeroDesign to calculate the aerodynamic coefficients. The comparisons made, show that CACS has superior results compared with GA. It can find better design points in a fewer evaluations. This is consistent with the previous results in [15,18].

The organization of this paper is as follows: In section 2 the external configuration design problem is defined as an optimization problem. The optimization problem is defined both in continuous and discrete spaces to prepare the solution by CACS and the considered GA, respectively. In the following two sections GA and CACS are introduced. The numerical results are given in section 5. Finally a conclusion is made in section 6.

2 Problem Definition

In this paper, the aerodynamic shape optimization of a missile is investigated using two different stochastic optimization approaches. The first one is a Genetic Algorithm and the second one is Ant Colony Optimization. To make the results comparable with those of the previous works, the same problem as Ref. [13] is investigated in this paper. The objective function is defined as the summation of the normal force coefficients over a set of given flight conditions. It can be formulated as follows:

$$f = \sum_{i=1}^{64} (C_{N_i}) \qquad (1)$$

Each objective function was sampled over eight Mach numbers at eight angles of attack to form a sufficient aerodynamic database. The Mach numbers are 0.3, 0.7, 0.9, 1.1, 1.5, 2.0, 3.0, and 4.0, and the angles of attack are 1.0, 2.0, 4.0, 6.0, 8.0, 12.0, 16.0 and 20.0. The Reynolds number is set to 6×10^6.

Next, the Engineering Code (EC) is used to calculate aerodynamic coefficients. This code can predict different aerodynamic coefficients at various flight conditions. The design parameters, passed by the optimization method to EC, are shown in Fig. 1. Note that though the nozzle is shown, it is merely for visualization purposes. The nozzle actually resides within the total length of the missile.

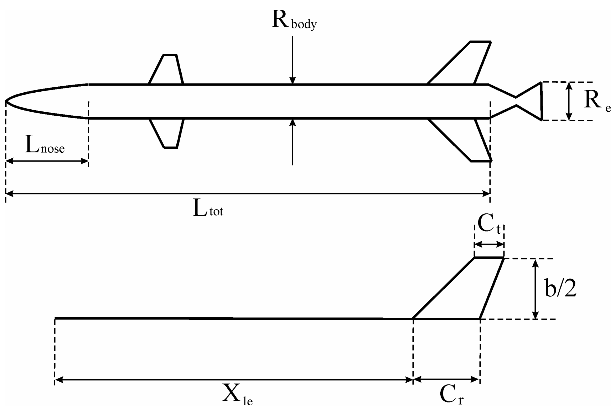
Each parameter is defined over a special range. In the case of discrete optimization methods, such as the GA Considered in this paper, the parameters are explored with a specific resolution. The parameters and their range and resolution are given in table 1. To avoid non-realistic configurations which cause some conflicts in design, three checks were made.

C1 - Tail control surfaces cannot be coincident with, or in front of the wing.

C2 - Tail control surfaces cannot overhang the aft end of the missile.

C3 - Wing cannot overhang nose portion.

These conflicts are checked before the objective function is evaluated. If any of these conflicts occur, a poor performance value will result. Therefore the optimization method will learn not to try these designs in future. So the optimization problem is to determine the parameters of table 1 such that the objective function (1) is maximized subject to the constraints C1 – C3.

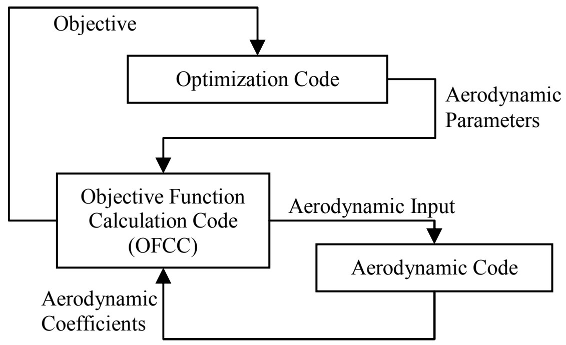


‘Fig. 1 Definition of the Aerodynamic parameters [13]

A high level description of the optimization process is shown in Fig. 2. The optimization code generates the aerodynamic parameters. These parameters are passed to the Objective Function Calculation Code (OFCC). The OFCC Calculates the objective function corresponding to each parameter set using EC. Finally OFCC passes the objective function to the optimization code.

Table 1 : Maximum, minimum and resolution of parameters governing external configuration design

Parameter	Minimum	Maximum	Resolution
Nose	0	1	1
Lnose	20	200	5
Ltot	300	700	5
Rbody	16	20	0.25
Rexit	4	30	0.25
Xlew	20	350	2.0
Bw	0	80.0	1.0
Crw	0	80.0	1.0
λtew	0.0	75.0	5.0
TRw	0	1.0	0.05
Xlet	350	680	2.0
Bt	0	80.0	1.0
Crt	0	80.0	1.0
λtet	0.0	75.0	5.0
TRt	0	1.0	0.05



'Fig. 2 General structure of the optimization process'

3 Genetic Algorithm

Genetic Algorithms (GAs) encode potential solutions of an optimization problem into chromosome like structures. Then allows these structures to compete, reproduce and mutate to create better solutions over generations. The same GA code as Ref. [13], called IMPROVE, is used in this paper. An additional improvement is also made by adding a creep operation as proposed in Ref. [13]. This operation can be described as follows: During reproduction, some of the parameters are randomly chosen. These parameters are limited to be changed only with the amount of their resolution.

The parameters, used to control the behavior of this variant of GA, include Crossover Probability, Mutation Probability, Creep Probability and Population Size. To make the results comparable with those of Ref. [13], the same setting of these parameters is utilized in this paper. The corresponding values are given in table 2.

Table 2 : GA Parameters for its behavior control

Parameter	Value
Crossover Probability	0.9 (90% of the times 2 survivors are mated they produce offspring
Mutation Probability	0.002 (2 out of every 1000 bits will mutate
Creep Probability	0.05 (5% of the parameters in a population will creep by their resolution
Population Size	250

The design variables must be converted into a binary form. The total number of genes corresponding to each chromosome can be calculated as follows:

$$N_g = \sum_{i=1}^{N_p} \left\{ Integer \left[\frac{Ln(\max(p_i) - \min(p_i))}{Ln(2) \times Resolution(p_i)} \right] + 1 \right\} \quad (2)$$

Where N_g is the total number of genes within each chromosome, N_p is the number of parameters to be optimized, and p_i is the i-th parameter $(1 \le i \le N_p)$. More details on the adopted GA can be found in Ref. [13].

4 Ant Colony Optimization

Ant algorithms were inspired by the observation of the real ant colonies. An important and interesting behavior of ant colonies is their foraging behavior, and in particular, how ants can find the shortest path without using visual cues. While walking from the food sources to the nest and vice versa, ants deposit on the ground a chemical substance called pheromone which makes a pheromone trail. Ants use pheromone trails as a medium to communicate with each other. They can smell pheromone and when they choose their way, they tend to choose paths with more pheromone. The pheromone trail allows the ants to find their way back to the food source or to the nest. Also, the other ants can use it to find the location of the food sources, which are previously found by their nest mates.

This pheromone trail following behavior can converge to the shortest path, once employed by a colony of ants. It means that, when there are more paths from the nest to a food source, a colony of ants may be able to use the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back.

Consider two different paths from the nest to the food source with different lengths. Initially there is no pheromone on the two branches, so ants select them with the same probability. Since the ants move at approximately constant speed, at each instant of time the number of ants who have passed the shorter path is greater than the number of ants who have not. Therefore when the ants start their return trip, more pheromone is present on the shorter path, increasing the probability of choosing it. Returning the ants through the shorter path refreshes it faster than the other one and compensates the pheromone evaporation. Thus in this way pheromone is accumulated on the shorter path and for the new ants who want to go to the food source, the probability of choosing it, will increase. Very soon all the ants will be using the shorter path.

4.1 Ant Colony System Basic Features

Ant Colony System (ACS) was one of the first algorithms proposed based on ACO. It was a discrete algorithm, and at first it was applied to the well-known Traveling Salesman Problem (TSP) [17], which is a discrete optimization problem. In this part we will shortly review the basic idea of ACS. Then in the subsequent part, the continuous version of ACS will be presented.

Consider a set of cities. TSP is defined as the problem of finding a minimal cost closed tour that visits all cities and each city only once. In a graph representation, the cities are the nodes and the connection lines between them are the edges. Each edge is associated with a cost measure, which determines the distance or cost of travel between two cities.

Ant colony system uses a graph representation, which is the same as that defined for TSP. In addition to the cost measure, each edge has also a desirability measure, called pheromone intensity, updated at run time by the ants.

Ant colony system works as follows: Each ant generates a complete tour by choosing the cities according to a probabilistic state transition rule. Ants prefer to move to cities, which are connected by short edges with a high amount of pheromone, while in some instances, their selection may be random.

Every time an ant in one city has to choose another city to move to, it samples a random number, q in [0,1]. If q becomes less than a given q_0, then the destination city is chosen by exploitation. It means that the one connected by the edge with the most ratio of pheromone intensity to distance, is chosen. Otherwise a city is chosen by exploration. In this case the one connected by the edge with the most ratio of pheromone intensity to distance, has the most chance to be chosen, but all other cities have also their chances to be chosen proportional to their ratio of pheromone intensity to distance.

While constructing a tour, ants also modify the amount of pheromone on the visited edges by applying a local updating rule. It concurrently simulates the evaporation of the previous pheromone and the accumulation of the new pheromone deposited by the ants when they are building their solutions.

Once all the ants have completed their tours, the amount of pheromone is modified again, by applying a global updating rule. Again a part of pheromone evaporates and all edges that belong to the globally best tour, receive additional pheromone conversely proportional to their length.

4.2 Continuous Ant Colony System Algorithm

A continuous optimization problem can be defined as the problem of finding the absolute maximum of a positive non-zero continuous cost function f(x), within a given interval [a,b], which the maximum occurs at a point x_s. In general f can be a multi-variable function, defined on a subset of R^n delimited by n intervals $[a_i, b_i]$, $i = 1, \ldots, n$.

The latest developed Continuous Ant Colony System (CACS) has all the major characteristics of ACS, but certainly in a continuous frame. These are a pheromone distribution over the search space which models the desirability of different regions for the ants, a state transition rule with both exploration and exploitation strategies, and a pheromone updating rule which concurrently simulates pheromone accumulation and pheromone evaporation.

4.2.1. Continuous Pheromone Model

Although pheromone distribution has been first modeled over discrete sets, like the edges of the traveling salesman problem, in the case of real ants, pheromone deposition occurs over a continuous space. Consider a food source, which is surrounded by several ants. The ants' aggregation around the food source causes the most pheromone intensity to occur at the food source position. Then increasing the distance of a sample point from the food source will decrease its pheromone intensity. CACS models this variation of pheromone intensity, in the form of a normal distribution function:

$$\tau(x) = e^{-\frac{(x - x_{\max})^2}{2\sigma^2}} \qquad (3)$$

Where $x_{\max}$ is the best point in the interval [a,b] which has been found from the beginning of the trial and σ can be interpreted as an index of the ants aggregation around the current maximum. Note that τ has been modeled as a Probability Distribution Function (PDF) which determines the probability of choosing each point x within the interval [a,b].

4.2.2. State Transition Rule

In CACS, pheromone intensity is modeled using a normal PDF, the center of which is the last best global solution and its variance depends on the aggregation of the promising areas around the best one. So it contains exploitation behavior. In the other hand, a normal PDF permits all points of the search space to be chosen, either close to or far from the current solution. So it also contains exploration behavior. It means that ants can use a random generator with a normal PDF as the state transition rule to choose the next point to move to.

4.2.3 Pheromone Update

Ants choose their destinations through the probabilistic strategy of equation (4). At the first iteration, there isn't any knowledge about the maximum point and the ants choose their destinations only by exploration. It means that they must use a high value of σ (associated with an arbitrary $x_{\max}$) to approximately model a uniform distribution positions.

During each iteration, pheromone distribution over the search space will be updated using the acquired knowledge of the evaluated points by the ants. This process gradually increases the exploitation behavior of the algorithm, while its exploration behavior will decrease. Pheromone updating can be stated as follows: The value of objective function is evaluated for the new selected points by the ants. Then, the best point found from the beginning of the trial is assigned to $x_{\max}$. Also the value of σ is updated based on the evaluated points during the last iteration and the aggregation of those points around $x_{\max}$. To satisfy simultaneously the fitness and aggregation criteria, a concept of weighted variance is defined as follows:

$$\sigma^2 = \frac{\sum_{j=1}^{k} (f_j - f_{\max})(x_j - x_{\max})^2}{\sum_{j=1}^{k} (f_j - f_{\max})}, \qquad (4)$$

$$for \quad all \quad j \quad in \quad which \quad f_j \neq f_{\max}$$

where k is the number of ants. This strategy means that the center of region discovered during the subsequent iteration is the last best point and the narrowness of its width is dependent on the aggregation of the other competitors around the best one. The closer the better solutions get (during the last iteration) to the best one, the smaller σ is assigned to the next iteration.

During each iteration, the height of pheromone distribution function increases with respect to the previous iteration and its narrowness decreases. So this strategy concurrently simulates pheromone accumulation over the promising regions and pheromone evaporation from the others, which are the two major characteristics of ACS pheromone updating rule.

5 Numerical Results

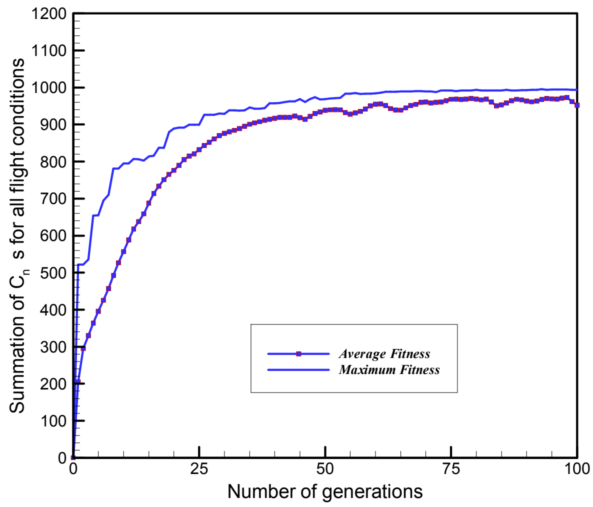
In this section the optimal solutions found using GA and CACS are compared. In both cases EC is used to calculate the aerodynamic coefficients. The obtained results are compared with those of GA+AeroDesign [13].

5.1 Optimization Results of GA

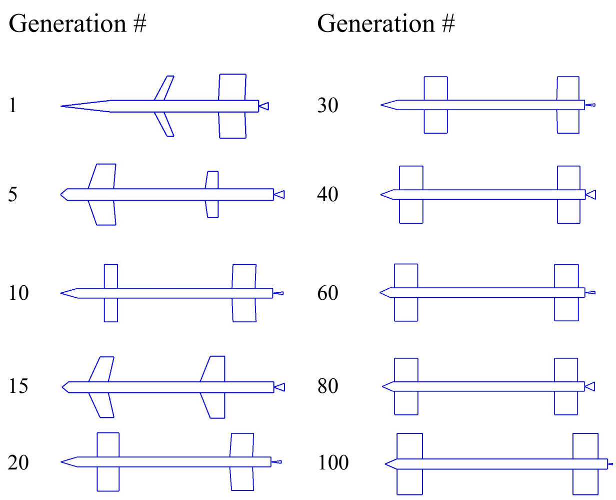
In this paper the available public domain version of IMPROVE code [19] was used and modified for GA. The parameters of GA are set according to table 2. Fig. 3 shows the history of the objective function. One can easily observe that the variation of objective function is negligible after 75 generations. Fig. 4 graphically shows the design history of the missile, ignoring the detail visualization of the nose shape. Though these design traits may seem strange at first, one must focus on wing-tail geometries, because the objective function has been defined as the summation of the normal force coefficients. It can be easily found from Fig. 4 that GA has successfully learned to make the wing and tail areas large and to minimize the body radius during the design process. The missile length increases during the process, which contributes to higher normal force coefficients. The nose length was also minimized to maximize the distance between two fin sets.

5.2 Optimization Results of CACS

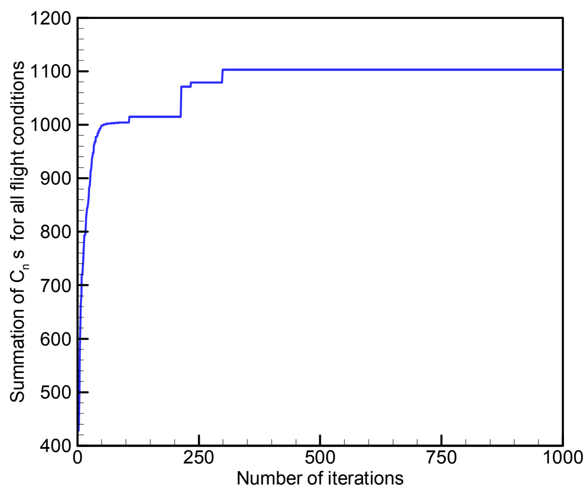
An important aspect of CACS with respect to the other optimization methods such as GA, is that it has only one control parameter, which is the number of ants (k). According to the previous experiences with CACS [15,18], the Value of k=10 has been found as a good selection that works well over a wide range of analytical and engineering optimization problems. The same setting is adopted in this paper. Fig. 5 shows the history of the objective function and Fig. 6 shows the graphical design history of the missile. A comparison between these results and those of GA shows a better performance for CACS rather than GA. The performance obtained from CACS after 1000 evaluations (100 iterations) is as good as the performance obtained from GA after 25000 evaluations (100 generations). Also the performance obtained from CACS after 5000 evaluations (500 iterations) is approximately 10 percent better than the performance obtained from GA after 25000 evaluations (100 generations).



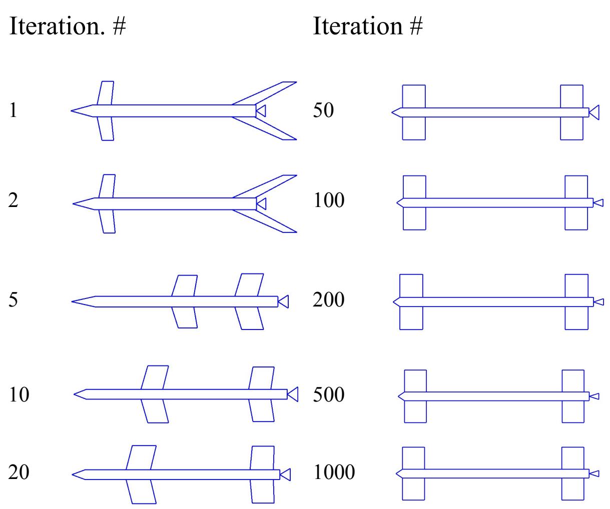
'Fig. 3 Variation of the objective function versus the number of generations, obtained from GA



'Fig. 4 External shape history obtained from the GA'



'Fig. 5 Variation of the objective function versus the number of iterations obtained from CACS



'Fig. 6 External shape history obtained from the CACS'

5-3 Comparison between MD and AeroDesign

In this section the results of GA+EC and CACS+EC, which are studied in this paper, are compared with those of GA+AeroDesign, obtained from Ref. [13]. Table 3 shows the optimal solutions found utilizing these three strategies. The results are consistent with each other. The parameters can be grouped into two categories. The first group include the parameters such as L_{total} . The results for these parameters are approximately the same in all cases, because they directly affect the value of the objective function. The second group include the parameters such as R_{exit} . Since these parameters have no effect on the objective function, the corresponding results are not necessarily the same.

A comparison between the performances of these three methods is made in table 4. These results also show a relatively good agreement between EC and AeroDesign. Table 4 also shows a better performance for CACS as compared to GA.

Table 3 : Optimal designs obtained from different strategies

Parameter	GA+MD	CACS+MD	GA+AeroDesign
Nose	Ogive	Ogive	unknown
L_{Nose}	35	23.92	20
L_{total}	700	699.98	700
R_{body}	16	16.002	16
R_{exit}	4.25	11.315	29.25
X_{lew}	38	23.92	44
b_w	80	79.998	80
C_{rw}	79	79.997	80
λ_{te_w}	0	0.0007	0
TR_w	1	0.9999	0.95
X_{let}	590	600.69	596
b_t	80	79.999	80
C_{rt}	79	79.992	80
λ_{te_t}	0	0.0002	0
TR_t	1	0.9999	1

Table 4 : Value of the objective function Compared between different strategies

Method	Number of evaluations	f
GA+AeroDesign	25000	960
GA+MD	25000	1000
CACS+MD	5000	1100

6 Conclusion

In this paper, the problem of aerodynamic shape optimization is investigated for unguided projectiles. Two stochastic optimization approaches, namely Genetic Algorithm (GA) and Continuous Ant Colony System (CACS), were applied to solve the problem. The later has been recently developed based on Ant Colony Optimization. The objective function was defined as the summation of normal force coefficients over a set of given flight conditions. An in-house developed Engineering Code (EC) was used to calculate the normal force coefficients over the flight conditions. The obtained results of CACS+EC were compared with those of GA+EC, as well as the results of a previous work (GA+AeroDesign). Comparisons show that CACS has superior results compared with GA. It can find better design points in a fewer evaluations. Also results show good agreements between EC and AeroDesign.

7 References

[1] Consentino, G., and Holst, T., "Numerical Optimization Design of Advanced Transonic Wing Configurations," AIAA Paper 85-0424, Jan. 1985.

[2] Kroo, I., and Takia, M., "A Quasi-Procedure, Knowledge-Based System for Aircraft Design," AIAA Paper 88-4428, Sept. 1988.

[3] Gage, P., and Kroo, I., "Development of the Quasi-Procedural Method for use in aircraft configuration Optimization," AIAA Paper 92 4693, Sept. 1992.

[4] Kroo, I., Altus, S., Braun, R., Gage, P., and Sobieski, I., "Multidisciplinary Optimization Methods for Aircraft Preliminary Design," AIAA Paper 94-4325, Sept. 1994.

[5] Tanrilulu, O., and Ercan, V., "Optimal External Configuration Design of Unguided Missiles," Journal of Spacecraft and Rockets, Vol. 35, No. 3, 1998, pp 312-316.

[6] Tavakoli, F., and Esfahanian, V., "External Configuration Optimization of Spinning Rockets Using Simplex Method," Proceedings of the first international conference of Iranian Aerospace Society, Tehran, Dec. 2000.

[7] Bramlete, M., and Cusic, R., "A Comparative Evaluation of Search Methods Applied to the Parametric Design of Aircraft," Proceedings of the third International Conference on Genetic Algorithms, CA, 1989, pp 213-218.

[8] Gage, P., and Kroo, I., "A Role of Genetic Algorithms in a Preliminary Design Environment," AIAA Paper 93-3933, Aug. 1993.

[9] Anderson, M. B., "Potential of Genetic Algorithms for Subsonic Wing Design," AIAA Paper 95-3925, Sept. 1995.

[10] Anderson, M. B., "Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design," AIAA Paper 96-4023, Sept. 1996.

[11] Selig, M. S., and Coverstone-Carroll, V. L., "Applications of a Genetic Algorithm to Wind Turbine Design," Journal of Energy Resources Technology, Vol. 118, March 1996, pp. 22-28.

[12] Cao, H. V., and Blom, G. A., "Navier-Stokes/Genetic Optimization of Multi-Element Airfoils," AIAA Paper 96-2487, 1996.

[13] Anderson, M. B., Burkhalter, J. E., and Jenkins, R. M., "Missile Aerodynamic Shape Optimization Using Genetic Algorithms," Journal of Spacecraft and Rockets, Vol. 37, No. 5, 2000, pp. 663-669.

[14] Xing, X. Q., and Damodaran, M., "Aerodynamic Shape Design of Nozzles Using A Hybrid Optimization Method", Proc. of the 4-th Singapore-MIT Alliance Annual Symposium, Singapore, 19-20 January 2004.

[15] Pourtakdoust, S. H., and Nobahari, H., "An Extension of Ant Colony System to Continuous Optimization Problems," Lecture Notes in Computer Science, Vol. 3172, 2004, pp. 294-301.

[16] Colorni, A., Dorigo, M. and Maniezzo, V., “Distributed Optimization by Ant Colonies,” *Proceedings of the First European Conference on Artifcial Life*, Elsevier Science Publisher, 1992, pp. 134–142.

[17] Dorigo M. and Gambardella, L. M., “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem,” *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, 1997, pp. 53–66

[18] Nobahari, H., and Pourtakdoust, S. H., "Optimization of Fuzzy Rule Bases Using Continuous Ant Colony System," Proceeding of the First International Conference on Modeling, Simulation and Applied Optimization, Sharjah, U.A.E., Paper# 243, 2005, 1-6.

[19] http://www.c-sharpcorner.com/Code/2002/July /GeneticAlgorithm.asp