JOURNAL OF AEROSPACE COMPUTING, INFORMATION, AND COMMUNICATION
Vol. 2, April 2005

Genetic Algorithms Applied to Multi-Objective Aerospace Shape Optimization

Terry L. Holst*
NASA Ames Research Center, Moffett Field, CA 94035

A genetic algorithm approach suitable for solving multi-objective optimization problems is evaluated using a series of aerodynamic shape optimization problems. The genetic algorithm is suitable for finding Pareto optimal solutions in search spaces that are defined by any number of genes and that contain any number of local extrema. A new masking array capability is included allowing any gene or gene subset to be eliminated as decision variables from the design space. This allows determination of the effect of a single gene or gene subset on the Pareto optimal solution. Results indicate that the genetic algorithm optimization approach is flexible in application and reliable.

Nomenclature

c_j	chord at j^{th} airfoil defining station
D_j	wing dihedral at j^{th} airfoil defining station
N_C	fixed number of chromosomes in each GA generation
N_G	number of genes in each chromosome
N_O	number of objectives
N_w	number of airfoil defining stations
p_1	user-specified probability controlling perturbation mutation operator ($0 \leq p_1 \leq 1$)
p_2	user-specified probability controlling mutation operator ($0 \leq p_2 \leq 1$)
$R(0, 1)$	random number generator returning a random value between 0.0 and 1.0
rle	airfoil leading edge radius of curvature (Sobieczky[27])
r_B	radius of base fuselage
r_S	radius of fuselage sting
t_j	airfoil thickness multiplier at j^{th} airfoil defining station
$x,\ y,\ z$	Cartesian coordinates centered at fuselage nose, x positive downstream, y positive to pilot's right, and z positive up
x_B	fuselage length
x_i	i^{th} gene or decision variable
x_N	length of fuselage elliptical nose
x_R, y_R, z_R	x, y, z position of wing root leading edge within fuselage coordinate system
x_T	length of fuselage boattail
$x_{\theta,j}$	center of twist at j^{th} airfoil defining station
xlo	x-location of lower-surface maximum airfoil thickness (Sobieczky[27])
xup	x-location of upper-surface maximum airfoil thickness (Sobieczky[27])
$x_{\max_i}$	user-specified maximum limit on i^{th} gene
$x_{\min_i}$	user-specified minimum limit on i^{th} gene

Received 18 August 2004; revision received 22 November 2004; accepted for publication 8 December 2004.

* Research Scientist, NASA Advanced Supercomputing Division, Applications Branch, Fellow AIAA.

y_j	y-location of j^{th} airfoil defining station
zlo	airfoil lower-surface maximum thickness (Sobieczky[27])
zte	position of trailing edge above $z = 0$ plane (Sobieczky[27])
zup	airfoil upper-surface maximum thickness (Sobieczky[27])
zxxlo	airfoil lower-surface curvature at $x =$ xlo (Sobieczky[27])
zxxup	airfoil upper-surface curvature at $x =$ xup (Sobieczky[27])
αte	angle between airfoil trailing edge bisector and $z = 0$ (Sobieczky[27])
β	user-specified parameter controlling size of perturbation mutations ($0 \leq \beta \leq 1$)
βte	airfoil trailing edge included angle (Sobieczky[27])
Δr	fuselage decision variable used with Hicks–Henne[28] bump functions
Λ_j	leading edge sweep at j^{th} airfoil defining station
θ_j	wing twist at j^{th} airfoil defining station

I. Introduction

NUMERICAL methods for optimizing the performance of engineering problems have been studied for many years. Perhaps the most widely used general approach involves the computation of sensitivity gradients. These methods—called gradient methods—have been utilized to produce optimal engineering performance in a wide variety of different forms. The reliability and success of gradient methods generally requires a smooth design space and the existence of only a single global extremum or an initial guess close enough to the global extremum to ensure proper convergence.

In contrast to gradient-based methods, design space search methods such as genetic algorithms (GA)—also referred to as evolutionary algorithms (EA)—offer an alternative approach with several attractive features. The basic idea associated with the GA approach is to search for optimal solutions using an analogy to the theory of evolution. The problem to be optimized is parameterized into a set of decision variables or genes. Each set of genes that fully defines one design is called an individual or a chromosome. A set of chromosomes is called a population or a generation. Each complete design or chromosome is evaluated using a "biological-like" fitness function that determines survivability of that particular chromosome. For example, in aerospace applications, the genes may be a series of geometric parameters associated with an aerospace vehicle that is to be optimized for payload delivered to orbit, aerodynamic performance or structural weight. The fitness function takes as input all the geometric parameters and returns a numerical value for the fitness—the size of the payload, the aerodynamic performance or the structural weight.

During solution advance (or "evolution" using GA terminology) each chromosome is ranked according to its fitness vector—one fitness value for each objective. The higher-ranking chromosomes are selected and thus continue to the next generation while the lower-ranking chromosomes are not selected. The newly selected chromosomes in the next generation are manipulated using various operators (combination, crossover or mutation) to create the final set of chromosomes for the new generation. These chromosomes are then evaluated for fitness and the process continues—iterating from generation to generation—until a suitable level of convergence is obtained or until a specified number of generations has been completed.

Constraints can easily be included in the GA optimization approach either by direct inclusion into the objective function definition—a so-called penalty constraint—or more commonly, by including one or more constraints into the fitness function evaluation procedure. For example, if a design violates a constraint, its fitness is set to zero (for maximization), that is, it does not survive to the next generation. Because GA optimization is not a gradient-based optimization technique, it does not need sensitivity derivatives. It theoretically works well in non-smooth design spaces containing several or perhaps many local extrema.

General GA details including descriptions of basic concepts can be found in Goldberg,[1] Davis,[2] and Beasley et al.[3,4] Additional useful studies, which survey recent activities in the area of GA research, including the presentation of model problems useful for evaluating GA performance, are given in Deb,[5] Van Veldhuizen and Lamont[6] and Jiménez et al.[7]

A disadvantage of the GA approach is expense. In general, the number of function evaluations required for a GA optimization to converge, exceeds the number of function evaluations required by a finite-difference-based gradient optimization (see the results presented in Obayashi and Tsukahara[8] and Bock[9]). This situation is offset, to an extent, by the ease with which GAs can be implemented in parallel or distributed computing environments.

GA search methods are particularly attractive for multi-objective or multidiscipline optimization, that is, optimization problems in which two or more objectives are simultaneously and independently optimized. These methods, referred to as MOGA (multi-objective genetic algorithm) methods, offer the ability to directly compute an approximation to the so-called "Pareto optimal set" in a single computation instead of the limited single design point traditionally provided by other methods.

The Pareto optimal set, or Pareto front, as it is commonly called, includes optimal solutions for each of the individual objectives, as well as a range of tradeoff solutions in between, which are themselves optimal solutions. Providing a range of solutions to a multi-objective optimization problem is a powerful approach because it allows the designer to see the effect of gene variation on the design space in the form of optimal tradeoffs. Thus, the designer can choose individual objective weighting factors after their full influence is quantitatively known.

As presented in the references above, MOGA performance has been evaluated for many model problems. This is important because many scenarios involving GA algorithm and parameter variations, as well as design space variations, can be studied quickly and efficiently. Ultimately, however, it is important to evaluate MOGA performance for more realistic engineering applications because a model problem will never contain all of an actual problem's design space intricacies. An interesting and difficult application for any numerical optimization approach is aerodynamic shape optimization, especially when the shapes are complex and the aerodynamics is nonlinear.

A number of studies have been performed that have utilized a GA approach for aerodynamic shape optimization. Some of these include airfoil optimization by Marco et al.,[10] Naujoks et al.,[11] Quagliarella and Della Cioppa,[12] Vicini and Quagliarella,[13] Hämäläinen et al.[14] and Epstein and Peigin,[15] missile aerodynamic shape optimization by Anderson et al.,[16] and wing optimization by Anderson and Gebert,[17] Sasaki et al.,[18] Oyama,[19] Ng et al.,[20] and Obayashi et al.[21] Additional examples in the area of turbomachinery optimization include rocket engine turbopump design by Oyama and Liou[22] and compressor blade design by Benni[23] and Oyama and Liou.[24]

Evaluation of MOGA performance for a complex geometry, nonlinear-aerodynamic application is the objective of this paper. This is achieved by utilizing a transonic wing-fuselage optimization problem with special emphasis upon the fuselage shape, especially near the wing-fuselage juncture. In the present implementation the wing vertical attachment point is utilized as a decision variable in the optimization process. This is a difficult multi-objective optimization problem—more so than many studied to date—because movement of the wing requires the simultaneous change of other fuselage decision variables in order to maintain a Pareto front solution near the Pareto front. Thus, the present problem represents a difficult and unique test for evaluating MOGA performance.

II. Problem Statement: Multi-Objective Optimization

The problem statement for optimization scenarios involving more that one objective, which are simultaneously optimized, is more difficult to state than for single-objective problems. This is because each objective must play a role in determining the optimal solution. In the optimization process, conflicts might arise among the various objective functions, that is, the optimal values of each individual objective, in general, will not occur for the same decision variable vector. As a result, the "optimal solution" for a multi-objective optimization problem is typically a range or a set of solutions, which represent an optimal set of tradeoffs in objective space.

To determine when one solution is better than another for multi-objective problems the concept of **dominance** is utilized.[1] A vector $\mathbf{U} = \mathbf{U}(u_1, \ldots, u_i, \ldots, u_N)$ is said to dominate another vector $\mathbf{V} = \mathbf{V}(v_1, \ldots, v_i, \ldots, v_N)$ if and only if $u_i \geq v_i$ for all i and there exists at least one value of i such that $u_i > v_i$. A vector defined on some domain Ω that is not dominated by any other vector defined on Ω is said to be **non-dominated** on Ω.

A multi-objective optimization problem can be stated as follows: Let $\mathbf{F}$ be a set of N_O scalar objective functions, f_k, each dependent upon the same decision variable vector $\mathbf{X}$, which is defined on some design space Ω

$$\mathbf{F} = \mathbf{F}(f_1(\mathbf{X}), \ldots, f_k(\mathbf{X}), \ldots, f_{N_O}(\mathbf{X})) \tag{1}$$

where the decision variable vector $\mathbf{X}$ consists of N_G independent components. The multi-objective optimization problem involves finding the set of $\mathbf{X} = \mathbf{X}^*$ that produce non-dominated values for $\mathbf{F} = \mathbf{F}^*$ on Ω. This set of values $\mathbf{F}^*$ is called the **Pareto optimal set** or the **Pareto front**.

III. Genetic Algorithm

The MOGA optimization procedure utilized to solve the multi-objective optimization problem described above is presented in Holst and Pulliam[25,26] and will only be summarized herein. This multi-objective optimization scheme uses real-number encoding to represent all design space decision variables or genes. Each set of genes that leads to the complete specification of an individual design is called a chromosome. The total number of genes in a chromosome is N_G. The total number of chromosomes in a generation is N_C. Both these quantities are user-specified parameters fixed in size at the beginning of optimization.

The GA begins with initialization, followed by fitness evaluation, ranking and selection—all standard features for MOGA optimization procedures. For selection, the design space is subdivided into a number of equal-sized regions or "bins." For the present study, involving two optimization objectives, 25 bins are utilized. That is, each objective along the evolving Pareto front is subdivided into five equal segments between the maximum and minimum Pareto front extremes. Thus, the entire two-dimensional objective space is divided into a total of $5 \times 5 = 25$ bins.

Each solution along the approximate Pareto front is placed into the appropriate bin according to the numerical values of its two objectives. Many of the design space bins will be empty because they lie beyond the Pareto front or because they occupy an interior portion of the design space. Some of the bins that contain solutions may have many entries—some may have few. This, of course, depends on how the evolving Pareto front is populated and how the Pareto front slices through the matrix of bins that have been created. The present selection process randomly chooses an equal number of chromosomes from bins that have entries. This promotes an equal distribution of entries along the evolving Pareto front and can enhance MOGA convergence. This selection algorithm is applicable for any number of objectives ($N_O \geq 2$).

After the new chromosomes have been selected and placed in a temporary holding array, they are modified using one of four different modification operators: (1) passthrough, (2) random average crossover, (3) perturbation mutation, and (4) global mutation. Each of these modification operators—described in detail in Holst and Pulliam[25,26]—is now briefly discussed.

Although the passthrough operator is called a modification operation, it does not modify the chromosome it operates on. As the name implies, passthrough simply passes the chromosome through to the next generation without modification. For all problems studied herein, there is one passthrough chromosome for each objective—two for the two-objective problems studied herein. The first passthrough chromosome carries the optimal value for objective one, the second for objective two. Use of passthrough in this way keeps the optimal values of the two objectives from decreasing (increasing) for maximization (minimization) problems and is sometimes called elitism in GA circles.

The random average crossover modification operator is implemented by selecting two random chromosomes from the newly selected generation. The two selected chromosomes are then averaged on a gene-by-gene basis to create the modified chromosome.

The perturbation mutation modification operator is implemented by selecting a random chromosome from the newly selected generation. Next, a probability test is performed for each gene x_i within the selected chromosome. This involves a call to the random number generator $R(0, 1)$, which returns a real number value between zero and one. If the returned random number is less than the user-specified probability p_1 the gene is modified using

$$x_i^{new} = x_i^{old} + (x_{\max_i} - x_{\min_i})[R(0,1) - 0.5]\beta \qquad (2)$$

where β is a user-specified tolerance that controls the size of the perturbation mutation (typically set to 0.1), and $x_{\max_i}$ and $x_{\min_i}$ are maximum and minimum gene constraints for the i^{th} gene. For sensible results the value of p_1 must be between 0 and 1.0. In the present study p_1 is always equal to 0.2. Because this operator can cause the value of a particular gene to violate one of its constraints ($x_{\max_i}$ or $x_{\min_i}$), checks are implemented to prevent this.

The mutation or global mutation modification operator is implemented similarly to the perturbation mutation operator. First, a random chromosome is chosen from the newly selected generation. Next, a probability test is performed for each gene. If the returned random number is less than p_2 the gene is given a completely different value using

$$x_i^{new} = (x_{\max_i} - x_{\min_i})R(0,1) + x_{\min_i} \qquad (3)$$

The parameter p_2 is a user-specified control parameter controling the number of genes that are modified. For sensible results p_2 must be between 0.0 and 1.0. In the present study p_2 is always equal to 0.2.

After the passthrough chromosomes are selected and "modified"—there are always two passthroughs for the present two-objective problems considered herein—the remaining chromosome modifications are evenly distributed—as closely as possible—among the remaining three modification operators.

IV. Geometry Parameterization

In aerodynamic shape optimization the geometric parameterization is always an important step because it effectively connects the aerodynamic analysis routine to the optimization routine. An analytic parameterization suitable for wing-fuselage configurations that are typical of the transonic flow regime is now described. Most of the parameters have common-sense definitions in which the name itself provides the definition, for example, wing chord, wing leading edge sweep or fuselage length.

A Cartesian coordinate system (x, y, z) is used for all configurations. The coordinate system origin is at the fuselage nose. A plane of symmetry $(y = 0)$ is inherently assumed for all configurations. The x coordinate is aligned with the freestream direction and is positive downstream; y is normal to the symmetry plane, positive to the pilot's right; and z is vertical, positive up. All length parameters are nondimensionalized using the wing root chord, that is, the wing chord length where it intersects the plane of symmetry.

A. Wing Parameterization

Wing geometries are parameterized using N_w airfoil defining sections or stations, each using the geometric parameterization of Sobieczky.[27] This parameterization was previously used by Oyama[19] for transonic wing optimizations in conjunction with a genetic algorithm.

In the present implementation the first defining station is always at the wing root and the last is always at the wing tip. Each defining airfoil section is characterized by ten parameters. These parameters along with brief definitions are listed in Table 1. A graphical description of these parameters is presented in Fig. 1.

Once these parameters are specified the airfoil coordinates (z as a function of x) are determined using a polynomial of the form

$$z = \sum_{n=1}^{6} a_n x^{n-1/2} \tag{4}$$

where the coefficients a_n are computed from the ten airfoil defining parameters. Two applications of Eq. (4) are required for a complete airfoil specification—one for the upper-surface coordinates and one for the lower-surface coordinates. Six simultaneous linear equations involving rle, xup, zup, zxxup, αte + βte/2 and zte are solved for the upper-surface coordinates, and six simultaneous linear equations involving rle, xlo, zlo, zxxlo, αte − βte/2, and zte are solved for the lower-surface coordinates. Because rle and zte are used for both surfaces, slope continuity at the leading edge and zero thickness at the trailing edge are always maintained.

Table 1 Definitions for airfoil section and wing planform parameters. A value for each parameter is required at each of N_w wing defining stations, $j = 1, \ldots, N_w$.

Symbol	Definition	Symbol	Definition
Airfoil section parameters		Planform parameters	
rle$_j$	Leading edge radius of curvature	c_j	Airfoil chord length
xup$_j$	x-location of upper-surface max thickness	D_j	Dihedral angle (deg)
zup$_j$	Upper-surface max thickness	Λ_j	Leading edge sweep (deg)
zxxup$_j$	Upper-surface curvature at x = xup$_j$	t_j	Airfoil section thickness multiplier
xlo$_j$	x-location of lower-surface max thickness	θ_j	Wing twist (deg)
zlo$_j$	Lower-surface max thickness	$x_{\theta,j}$	Center of twist
zxxlo$_j$	Lower-surface curvature at x = xlo$_j$	y_j	Spanwise defining station location
zte$_j$	Position of trailing edge above $z = 0$		
αte$_j$	Angle between trailing edge bisector and $z = 0$ plane (deg)		
βte$_j$	Trailing edge included angle (deg)		

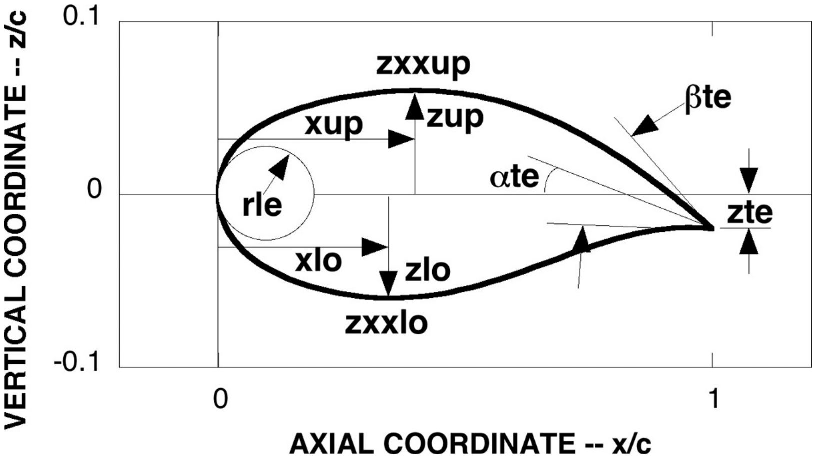


Fig. 1 Airfoil parameterization used for each wing defining station.[27]

Once the airfoil coordinates are constructed for each defining station, wing coordinates are constructed using linear lofting and then modified according to a set of planform parameters. Seven parameters are utilized at each planform defining station, producing a total of 7 × N_w planform parameters. These parameters are also listed in Table 1 along with brief definitions.

Certain geometric parameters have predetermined values or are not formerly used in the wing definition process. For example, the root chord length is always unity, and the spanwise location for the root airfoil defining station is always zero. The tip airfoil defining station values for the dihedral angle and the leading edge sweep angle have no meaning, but are nevertheless included in the list for each wing parameterization.

B. Fuselage Parameterization

The first step in the fuselage parameterization is to define a base fuselage, which is analytically constructed in three sections. The nose section is an ellipse of revolution. The main body section is a right circular cylinder that smoothly fairs into the nose section. And the boattail section is a sine curve of revolution smoothly connecting the main portion of the fuselage with a downstream sting. The sting—not formally part of the fuselage—is a small-radius right circular cylinder that extends to the downstream outflow boundary. A total of five parameters are required to specify this base fuselage. Three additional parameters—the three coordinates that indicate where the wing intersects the fuselage (x_R, y_R, z_R)—are required to complete the base wing-fuselage geometry definition. These eight parameters are listed and defined in Table 2. Note: In the present implementation y_R—like y_1 in Table 1—is inherently assumed to be zero. That is, the wing root is always aligned with the $y = 0$ plane of symmetry. Nevertheless, y_R is included in the Table 2 for completeness.

Modification of the base fuselage shape is achieved using a series of Hicks–Henne bump functions.[28] A total of N_x bump functions are distributed axially with equal spacing along the x-direction for each of N_T circumferential

Table 2 Base fuselage parameter definitions.

Symbol	Definition
x_R	x-location of wing root leading edge
y_R	y-location of wing root leading edge
z_R	z-location of wing root leading edge
x_B	Body length
x_N	Length of ellipsoid nose
r_B	Radius of base fuselage cylinder
x_T	Length of fuselage boattail
r_S	Radius of fuselage sting

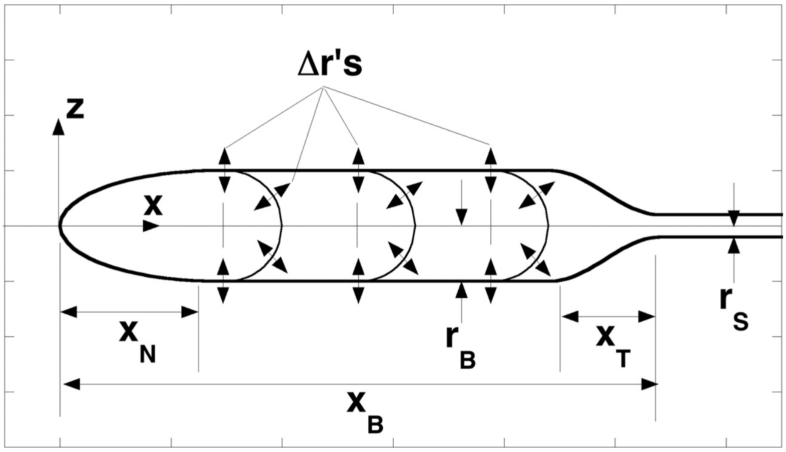


Fig. 2 Sketch showing decision variable definitions for fuselage parameterization.

stations, also distributed with equal spacing in the circumferential direction. This allows a total of $N_x \times N_T$ decision variables that are used to produce incremental radius perturbations to the base fuselage, which are then converted to surface grid coordinates for volume grid generation.

The Δr decision variables represent maximum amplitudes that are used in the Hicks–Henne[28] sinusoidal bump functions to produce a smooth distribution of radial increments over the entire fuselage. Depending upon how the Δr quantities are constrained in the MOGA process, the local radius of the fuselage can be increased or decreased (thickened or thinned) relative to the base fuselage. The logic for this is much the same as in the original approach by Hicks and Henne for wing surfaces. In the present implementation, the "flat" wing surface containing an array of bump functions is effectively wrapped around the fuselage. Instead of producing a blended array of z-coordinate perturbations for the wing, the present approach produces a blended array of radial perturbations for the fuselage.

Any position on the base fuselage, except the nose $x = 0$ or tail $x = x_B$, can be changed using this approach. The changes from one grid point to the next are always smooth because of the blending nature of the Hicks–Henne bump functions. A sketch showing definitions for all fuselage parameters, including one possible arrangement of Δr decision variables involving three axial stations ($N_x = 3$) and four circumferential stations ($N_T = 4$), is given in Fig. 2.

The baseline wing-fuselage geometry defined above does not have experimental data. As such, no experimental/computational comparisons are presented herein. Instead, comparisons between different solutions along the Pareto front, involving geometric and flow field variations, will be the primary mechanism for establishing performance of the present approach.

C. Masking Array

Associated with each decision variable, that is, each gene, is an integer value—either one or zero—stored in a masking array called "mask." The masking array is established at the beginning of the optimization computation and tells the MOGA which parameters are available for modification in the optimization process—using the crossover and mutation operators—and which to leave unmodified. For example, the mask values for c_1, y_1, Λ_{N_w}, D_{N_w} and y_R are always zero. These parameters, as described above, are carried along with each chromosome but are never used by the MOGA process. The masking array can also be used to limit the size of the design space. Only those genes with mask $= 1$ are modified in the search for optimal problem objectives.

V. Computed Results

Computed results for a two-objective optimization problem involving a wing-fuselage configuration flying in the transonic flow regime—$M_\infty = 0.84$—are presented next. The two objectives—lift-to-drag ratio at fixed lift and the configuration's volume—are simultaneously maximized by the previously presented MOGA optimization procedure. The aerodynamic lift-to-drag ratio is evaluated for each candidate design using the TOPS full potential solver[29] by iterating on angle of attack to obtain the lift-to-drag ratio at the specified value of lift—$C_L = 0.45$. Each TOPS solution requires about 2 minutes of CPU time on a 600 MHz processor. The tolerance on the lift iteration utilized throughout this study is ±1%, but in most cases the error is much less. See Holst[29] for grid generation and flow solver details associated with the TOPS code.

The wing-fuselage volume computation utilizes a straightforward algorithm that divides the fuselage into a series of cross-sections, computes the area of each cross-section using a simple triangularization and then computes the volume by marching along the fuselage using trapezoid integration. The wing volume is computed using a similar algorithm that is deactivated for those wing cross-sections that lie inside the fuselage.

There are several reasons lift-to-drag ratio (at fixed lift) and wing-fuselage volume have been chosen as the two objective functions for the present optimization problem. First, there is a clear-cut conflict between these two objectives. When the volume is maximized the lift-to-drag ratio suffers and vice versa. Thus, a classical engineering trade-off problem exists. How the MOGA performs on such a real-world problem is the key aspect of this study, not the development of new aeronautical engineering concepts.

In addition, the vehicle volume was chosen as an objective because it is an easy quantity to understand. The direct computation of maximum volume is easy to perform, thus providing a simple check for the results obtained by the MOGA—at least for one extreme on the Pareto front. Lastly, this problem is interesting in that it is complicated by transonic-flow shock-wave-induced nonlinearity and by a large number of decision variables that have large sensitivity variations.

The present wing-fuselage parameterization with two wing defining stations, $N_w = 2$, consists of 66 parameters. Airfoil defining station one is at the wing root (symmetry plane), and station two is at the wing tip. There are 24 Δr values used to modify the baseline fuselage—six equally-spaced axial stations, each with four equally-spaced circumferential locations. The axial positions are located at $x/c = 0.71$, 1.43, 2.14, 2.86, 3.57, and 4.29, where the fuselage length is fixed at $x_B = 5.0$. The circumferential positions are located at $\varphi = -90°$, $-30°$, $30°$, $90°$, where $\varphi = -90°$ corresponds to the fuselage keel line, and $\varphi = 90°$ corresponds to the crown line.

Maximum and minimum constraint values for each of the geometric parameters, as well as masking array values, are given in Table 3. As can be seen, 43 of the 66 genes that make up each chromosome have masking array values of one, and 23 have values of zero, that is, only 43 genes are actually modified during the optimization process—$N_G = 43$. Note also that the max-min values associated with each of the Δr values are not symmetric about zero. Along the keel and crown lines, for example, any Δr value averaged between the maximum and minimum constraints, will be positive, and along the other circumferential stations (along the fuselage side), the same averaging will produce negative values. The maximum and minimum Δr constraints were chosen in this way to keep the vertical extent of the fuselage from becoming too small. A difficulty arises in the wing-fuselage line of intersection computation if the intersection line contacts the symmetry plane. This results in fuselage designs that have depth-to-width ratios that generally exceed unity.

A UNIX script was written to allow parallel implementation of the MOGA routine using any number of chromosomes. One function evaluation was performed per processor. In particular, the first wing-fuselage computation utilized 34 chromosomes ($N_C = 34$). With two passthrough chromosomes this leaves 32 modified chromosomes requiring function evaluations during each generation, which are conveniently mapped onto 32 processors. During initialization when 34 function evaluations are actually required, 32 are computed. The final two are obtained by duplicating two arbitrary results from the first set of 32 computations.

Development of the approximate Pareto front as a function of generation number (n) is displayed in Fig. 3 for two cases—a baseline case and a second case in which the mutation process was aggressively biased toward the design space boundary. For the baseline case each mutated gene value was chosen to lie between that particular gene's maximum and minimum constraints using a random number generator, as presented in Eqs. (2) and (3).

For the second case—hereafter called the aggressive mutation case—the perturbation and global mutation operators were modified. Each new gene value was randomly chosen from a much larger range—an order of magnitude larger than the range established by the user-specified constraints. If the resulting gene value fell between the original constraints for that gene, that value was used without modification. If it was greater than the maximum constraint or less than the minimum constraint, it was reset to the nearest constraint limit. All other aspects of the MOGA mutation operators as outlined above [Eqs. (2) and (3)] were unaltered.

Table 3 Maximum and minimum gene constraints with masking array (m) values.

gene	max	min	m	gene	max	min	m	gene	max	min	m	gene	Max	min	m
Defining airfoil section 1				Defining airfoil section 2				Base fuselage parameters							
rle$_1$	0.014	0.014	0	rle$_2$	0.014	0.014	0	x_R	1.5	1.5	0	x_N	1.5	0.5	1
xup$_1$	0.45	0.35	1	xup$_2$	0.45	0.35	1	y_R	0.0	0.0	0	r_B	0.35	0.35	0
zup$_1$	0.08	0.05	1	zup$_2$	0.08	0.05	1	z_R	0.15	−0.15	1	x_T	1.5	0.5	1
zxxup$_1$	−0.2	−0.8	1	zxxup$_2$	−0.2	−0.8	1	x_B	5.0	5.0	0	r_S	0.1	0.1	0
xlo$_1$	0.45	0.35	1	xlo$_2$	0.45	0.35	1	Fuselage perturbations							
zlo$_1$	−0.05	−0.08	1	zlo$_2$	−0.05	−0.08	1	$\Delta r_{1,1}$	0.05	−0.02	1	$\Delta r_{1,3}$	0.02	−0.05	1
zxxlo$_1$	0.8	0.2	1	zxxlo$_2$	0.8	0.2	1	$\Delta r_{2,1}$	0.05	−0.02	1	$\Delta r_{2,3}$	0.02	−0.05	1
zte$_1$	0.0	0.0	0	zte$_2$	0.0	0.0	0	$\Delta r_{3,1}$	0.05	−0.02	1	$\Delta r_{3,3}$	0.02	−0.05	1
αte$_1$	8.0	0.0	1	αte$_2$	8.0	0.0	1	$\Delta r_{4,1}$	0.05	−0.02	1	$\Delta r_{4,3}$	0.02	−0.05	1
βte$_1$	10.0	10.0	0	βte$_2$	10.0	10.0	0	$\Delta r_{5,1}$	0.05	−0.02	1	$\Delta r_{5,3}$	0.02	−0.05	1
Planform defining station 1				Planform defining station 2				$\Delta r_{6,1}$	0.05	−0.02	1	$\Delta r_{6,3}$	0.02	−0.05	1
c_1	1.0	1.0	0	c_2	0.5	0.5	0	$\Delta r_{1,2}$	0.02	−0.05	1	$\Delta r_{1,4}$	0.05	−0.02	1
D_1	0.0	0.0	0	D_2	0.0	0.0	0	$\Delta r_{2,2}$	0.02	−0.05	1	$\Delta r_{2,4}$	0.05	−0.02	1
Λ_1	37.0	37.0	0	Λ_2	37.0	37.0	0	$\Delta r_{3,2}$	0.02	−0.05	1	$\Delta r_{3,4}$	0.05	−0.02	1
t_1	1.0	1.0	0	t_2	1.0	1.0	0	$\Delta r_{4,2}$	0.02	−0.05	1	$\Delta r_{4,4}$	0.05	−0.02	1
θ_1	3.0	0.0	1	θ_2	0.0	−3.0	1	$\Delta r_{5,2}$	0.02	−0.05	1	$\Delta r_{5,4}$	0.05	−0.02	1
$x_{\theta,1}$	0.5	0.5	0	$x_{\theta,2}$	0.5	0.5	0	$\Delta r_{6,2}$	0.02	−0.05	1	$\Delta r_{6,4}$	0.05	−0.02	1
y_1	0.0	0.0	0	y_2	2.4	2.4	0								

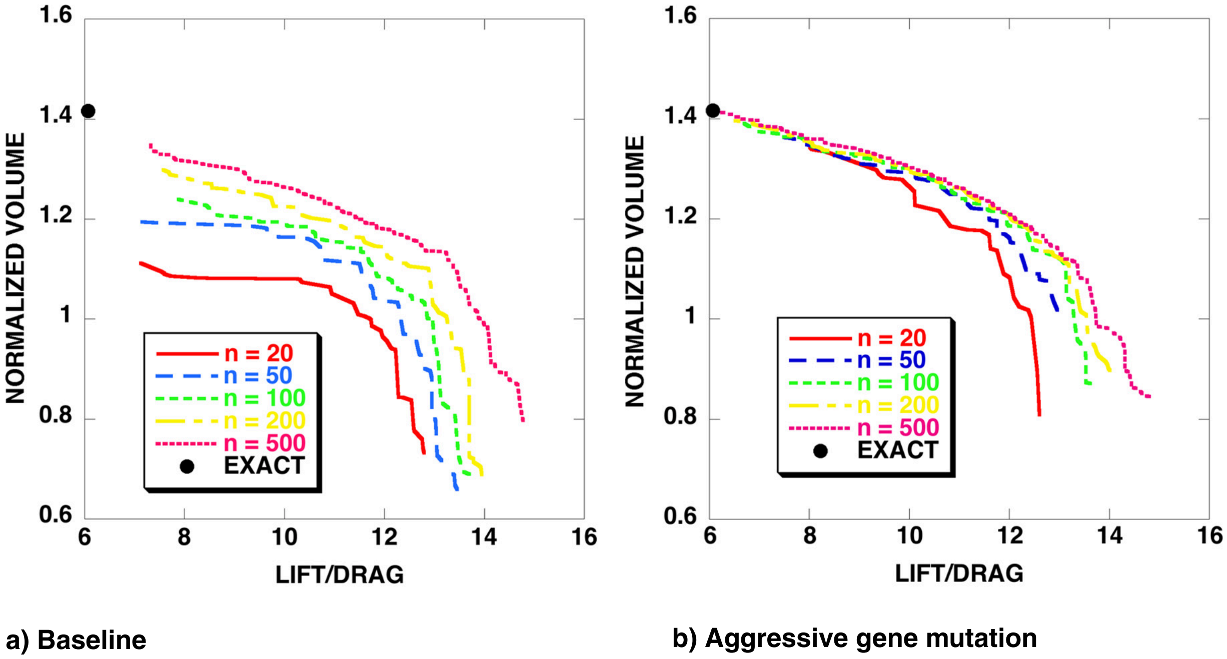


Fig. 3 Pareto front convergence for the wing-body optimization involving drag minimization and volume maximization, $C_L = 0.45$, $M_\infty = 0.84$, $N_C = 34$, $N_G = 43$.

Use of aggressive mutation increases the likelihood that a number of gene values in any given MOGA generation will end up at a gene constraint limit, but not as dramatically as one might expect. For example, for the global mutation operator, responsible for 33% of all gene modifications (approximately), a modified gene at a gene constraint limit is produced 6% ($\sim 0.33 \times 0.2 \times 0.9$) of the time, a modified gene at an interior value is produced 0.7% ($\sim 0.33 \times 0.2 \times 0.1$) of the time, and the original gene value is retained 26% ($\sim 0.33 \times 0.8$) of the time.

For the perturbation mutation operator, also responsible for 33% of all gene modifications, it is more difficult to estimate how many gene values are selected at the constraint limits using aggressive mutation because this modification operator depends on the initial gene value [the x_i^{old} value in Eq. (2)]. Assuming a uniform distribution of x_i^{old} values within the $x_{\max_i} - x_{\min_i}$ range, $\beta = 0.1$ and $p_1 = 0.2$, 5% ($\sim 0.33 \times 0.2 \times 0.75$) of the genes will be assigned interior values, 1.6% ($\sim 0.33 \times 0.2 \times 0.25$) will be assigned a constraint value and 26% ($\sim 0.33 \times 0.8$) will not be modified.

The random average crossover modification operation, also responsible for 33% of all gene modifications, is the same for the baseline and aggressive mutation cases. Each modified gene takes on an interior value except in those cases when the two original gene values being averaged both exist at the same constraint limit.

Development of the aggressive mutation approach was motivated by realizing that the maximum-volume Pareto extreme for the present problem lies on the gene space boundary. In particular, 38 of the 43 modifiable genes are maximized or minimized by the MOGA in achieving the maximum-volume solution. In addition, the minimum-drag point (at fixed lift) is achieved with 27 maximum or minimum gene values. Thus, a mutation process that biases toward the design space boundary seems to be logical and, for the present case, as seen by comparing Figs. 3a and 3b, produces a significant acceleration in MOGA convergence.

For the aggressive mutation case 96% of the normalized maximum volume—which occurs at a value of 1.42—is achieved after 50 generations. The baseline MOGA solution, after 50 generations, has achieved only 85% of the maximum volume. The minimum-drag point is not significantly affected by the mutation strategy, converging at about the same rate for both approaches. The convergence characteristics, displayed in Fig. 3 for one solution, were generally observed for every solution computed for the present wing-fuselage problem.

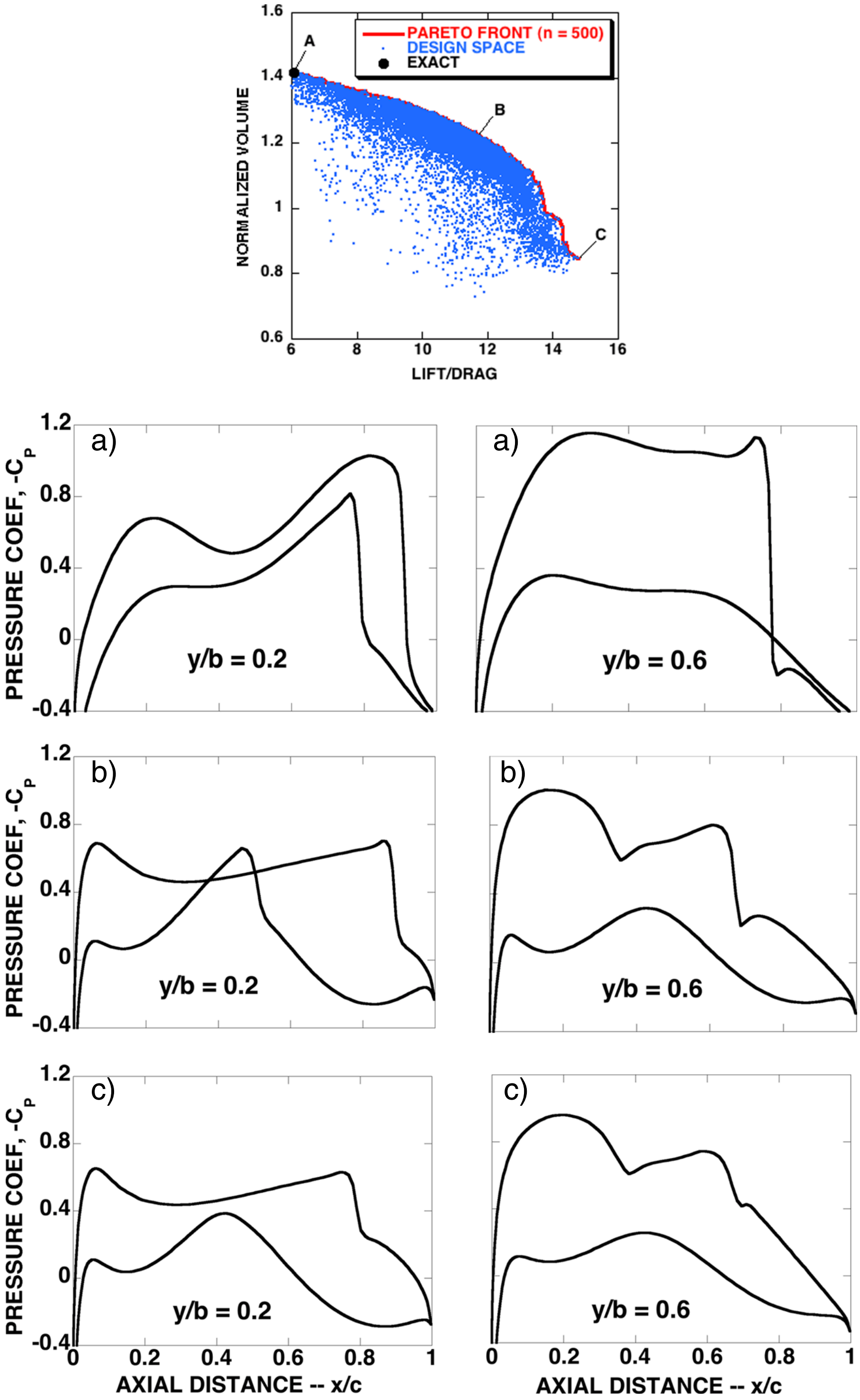


Fig. 4 Design space for the wing-body two-point optimization problem, $C_L = 0.45$, $M_\infty = 0.84$, $N_C = 34$, $N_G = 43$. Wing sectional pressure distributions presented at three positions along approximate Pareto front, a) max volume, b) intermediate, c) min drag.

For multi-objective optimization problems in which the Pareto front does not lie along the gene-space boundary, the aggressive mutation approach may not be advantageous. More research is needed to answer this question. All additional cases presented herein utilize the aggressive mutation approach.

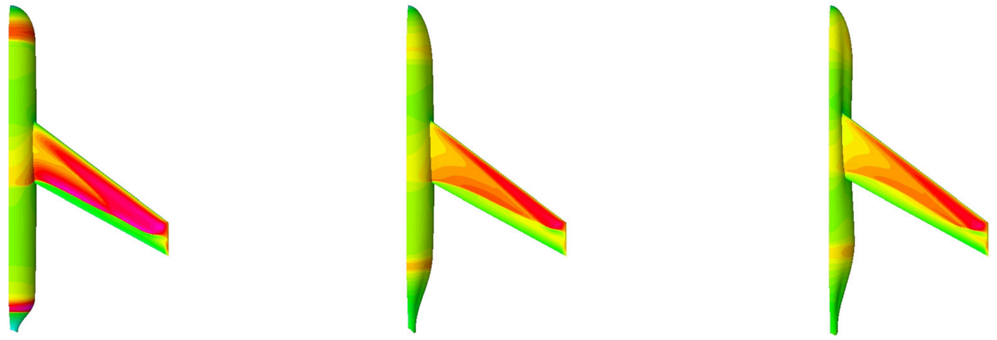
Several physical characteristics associated with the present wing-fuselage design space are presented in Fig. 4. The approximate Pareto front, identical to the $n = 500$ curve from Fig. 3b, is duplicated as a solid line. Each dot lying below and to the left of this curve represents one function evaluation (one call to the TOPS analysis code). The three positions called out along the approximate Pareto front—A, B, and C—indicate the design-space locations for the three pressure distributions plotted in the lower portion of Fig. 4. They correspond to solutions at A) maximum volume, B) intermediate, and C) minimum drag (at fixed lift).

For each solution two airfoil pressure distributions are presented. The first is inboard near the wing-fuselage juncture ($y/b = 0.2$), and the second is near the wing mid-semi-span ($y/b = 0.6$). The most obvious feature from these results is the change in shock strength as the approximate Pareto front is traversed. The maximum-volume solution (A) contains strong shocks, on both the upper- and lower-wing surfaces, while the minimum-drag solution (C) contains mild shocks on only the upper-wing surface.

Upper-surface Mach number contours for each of the three solutions presented in Fig. 4 are displayed in Figs. 5. This series of figures shows the shock wave pattern variation on the upper surface of the wing-fuselage configuration as the solution transitions from the maximum-volume Pareto front extreme to the minimum-drag extreme. For the maximum-volume extreme there are strong shock waves, not only on the wing, as seen in Fig. 4, but also on the fuselage near the nose and tail. The fuselage shock waves are caused by the MOGA's selection of minimum values for the x_N and x_T genes. These values help to maximize the vehicle volume, but cause shock waves to form at both the fuselage nose and tail due to excessive over-expansion.

As the solution propagates along the approximate Pareto front, trading volume to obtain a reduction in drag, the most noticeable initial change is associated with the fuselage nose and tail fairings. By choosing larger values for x_N and x_T, the drag is significantly decreased without sacrificing a large amount of volume. This can be seen by examining the intermediate solution in Fig. 5 and noting its position on the Pareto front in Fig. 4. Further reductions in vehicle volume to achieve improvements in drag are achieved via the complex task of area ruling the fuselage in the vicinity of the wing-fuselage juncture. This situation can be observed by looking at the minimum-drag solution in Fig. 5. More on this, including the comparison of selected fuselage cross-sections will be presented subsequently.

Pressure distributions along the fuselage for the three solutions described in Figs. 4 and 5 are presented in Figs. 6. Results are plotted along three fuselage circumferential stations, the keel line ($\varphi = -90°$), the side, ($\varphi = 0°$), and the crown line ($\varphi = 90°$). Each of these solutions involves a high-mounted wing configuration, which is chosen by the MOGA optimizer regardless of the solution's position along the computed Pareto front. More will be presented on this aspect subsequently. Because of this the side fuselage pressure distribution ($\varphi = 0°$) lies below the wing for each

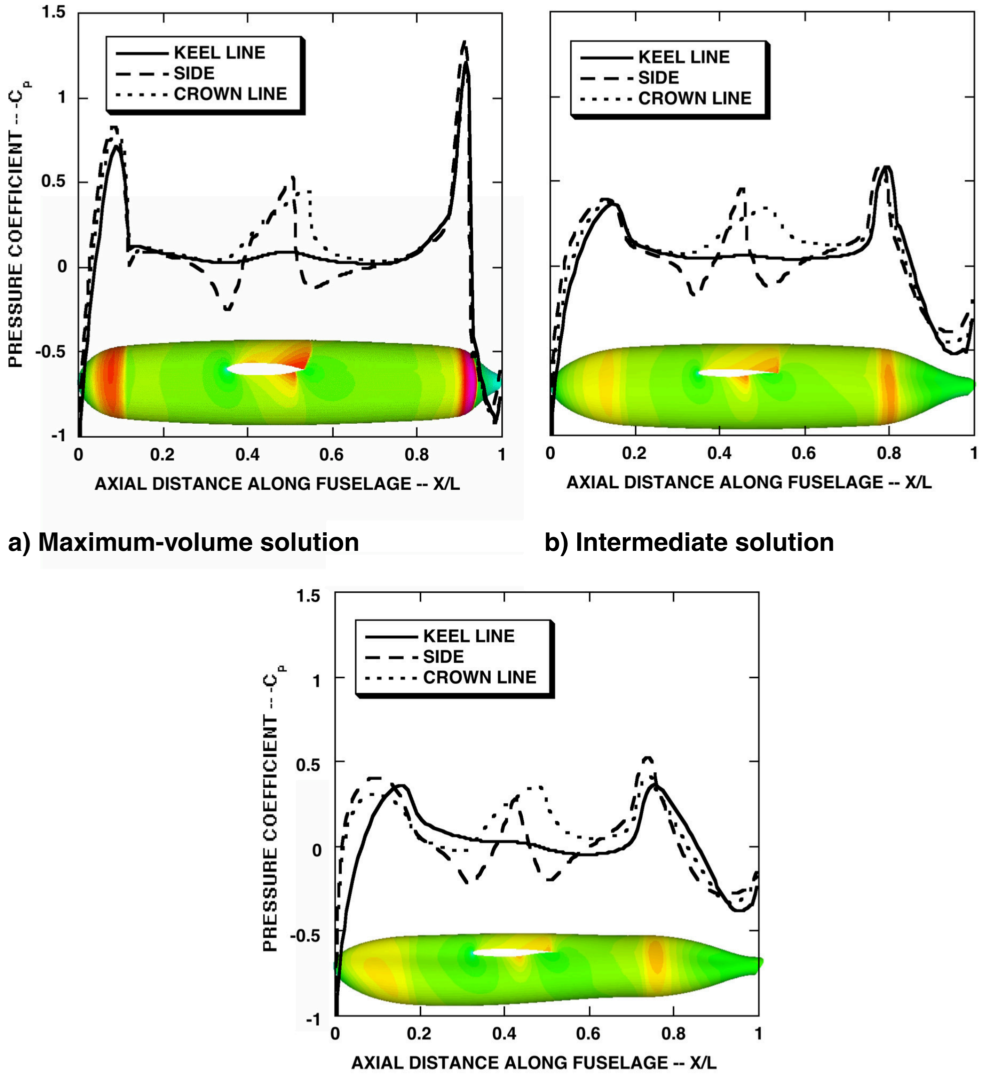


a) Maximum-volume solution

b) Intermediate solution

c) Minimum-drag solution

Fig. 5 Surface Mach number distributions for the wing-fuselage optimization problem presented in Fig. 4 at three positions along the approximate Pareto front.



a) Maximum-volume solution

b) Intermediate solution

c) Minimum-drag solution

Fig. 6 Fuselage pressure distributions for the problem presented in Figs. 4 and 5 at three positions along the Pareto front. Contour plots show Mach number distributions.

of these three solutions. In addition, Mach number contours as viewed from the fuselage's side are also displayed in Figs. 6.

As can be seen from Fig. 6a (and previously discussed in conjunction with Fig. 5) the expansions at the fuselage nose and tail are quite large and lead to strong shock waves. As the fairings at the nose and tail are improved, see Figs. 6b and c, the expansions and resulting shocks are greatly reduced. The shock wave induced on the fuselage by wing-fuselage interference can also be seen in Fig. 6a. Thinning the wing, as well as adding wing aft camber, reduces this effect in the intermediate solution (Fig. 6b). Fuselage area ruling reduces the wing-fuselage interference shock strength even further as the minimum-drag point is approached (Fig. 6c).

The shape of the fuselage can be seen in more detail in Fig. 7, where several fuselage cross-sections are presented for the problem displayed in Figs. 4 and 5. For each cross-section there are two curves plotted. The solid curve corresponds to the solution obtained at the minimum-drag Pareto front extreme, and the dashed curve corresponds to the solution obtained at the maximum-volume extreme. For this solution the wing root leading edge is placed at

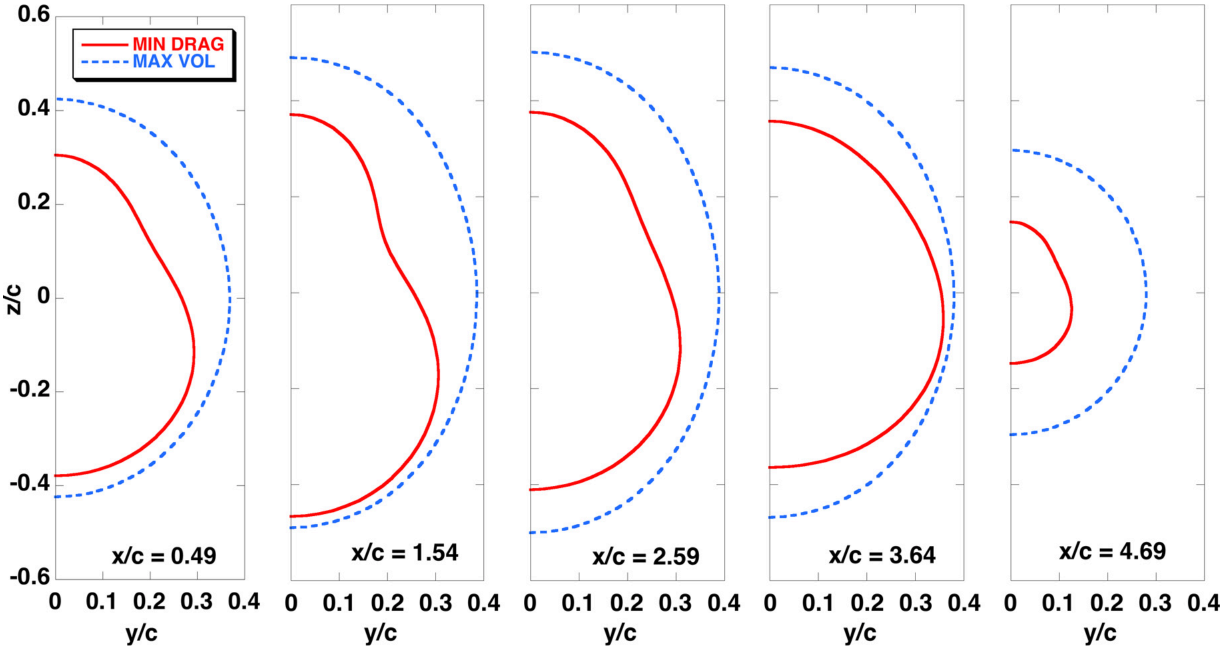


Fig. 7 Selected fuselage cross-sections for the wing-fuselage optimization problem of Figs. 4 and 5 showing differences between the two approximate Pareto front extremes.

$(x_R, y_R, z_R) = (1.5, 0.0, 0.15)$. Note that $x_R = 1.5$ and $y_R = 0.0$ are set at the beginning of the computation and are held fixed, and that $z_R = 0.15$ is the value selected by the MOGA optimization for this particular gene from the range $-0.15 \leq z_R \leq 0.15$ (see Table 3). Thus, the wing intersects the fuselage between stations 2 and 3 ($x/c = 1.54$ and 2.59) and at the highest position allowed by the z_R gene constraints. Note that the fuselage cross-sections are plotted before the wing-fuselage line of intersection is computed.

From the cross-sections displayed in Fig. 7—those associated with the minimum-drag Pareto front extreme (solid curves)—it is easy to see how the fuselage was modified or "area ruled" in the vicinity of the wing-fuselage juncture to obtain the final amount of drag reduction. Volume is subtracted along the upper portion of the fuselage near the wing intersection ($x/c = 1.54$ and $x/c = 2.59$), creating "pear-shaped" cross-sections in this location. Every cross-section from the maximum-volume solution contains more area than the corresponding minimum-drag cross-section, as each Δr value for the maximum-volume solution is individually maximized.

Optimized wing-root cross-sections for the two Pareto front extremes presented in Figs. 4 and 5 are displayed in Fig. 8. Note that the vertical axis has been expanded to facilitate the comparison. These two airfoils are plotted in "rigged" coordinates, that is, the leading edge is at $x/c = 1.5$ and $z/c = 0.15$. The airfoil representing the minimum-drag Pareto front extreme (solid curve) possesses minimum thickness with a significant amount of aft camber, that is, it is a supercritical airfoil section. The airfoil representing the maximum-volume Pareto front extreme (dashed curve) possesses maximum thickness and is essentially symmetric. These characteristics generally persist across the entire wing.

In order to assess convergence efficiency it is important to establish a suitable error norm that will accurately measure the error as the optimization process converges. For multi-objective optimization problems this can be difficult because error typically varies along the computed Pareto front. Zitzler et al.,[30] Knowles and Corne[31] and Fonseca and Fleming[32] discuss this issue and present several error norm alternatives. In the present study, the area between the developing Pareto front and the exact Pareto front—or a tightly converged approximation to it—is computed using an area triangularization algorithm.[33] The resulting "Pareto front error" plotted versus the number of function evaluations provides an assessment of convergence efficiency—at least for problems with two objectives.

Convergence efficiency for the present wing-fuselage optimization problem is presented in Fig. 9 for three different population sizes, 34, 66 and 130 chromosomes. Each curve is averaged over two solutions in an attempt to eliminate

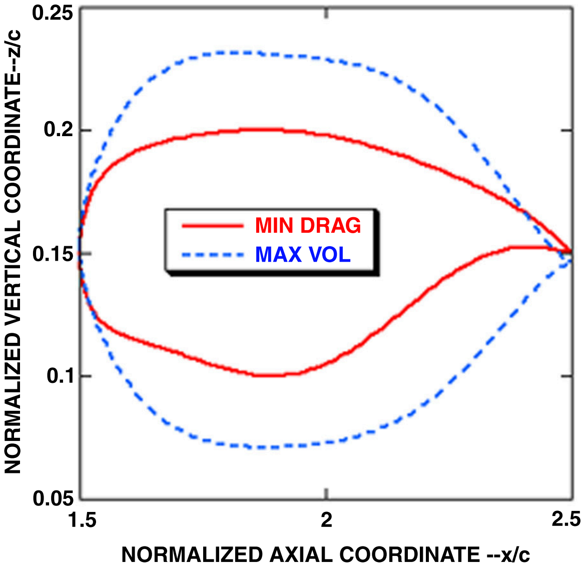


Fig. 8 Wing root airfoil cross-sections for the wing-fuselage optimization problem of Figs. 4 and 5 showing differences between Pareto front extremes. Note expanded vertical axis.

statistical variation. As can be seen each convergence history is nearly the same, indicating that MOGA convergence is not a function of population size. This result is compatible with the results presented in Holst and Pulliam[26] where several model problems were used to evaluate MOGA convergence.

Because the number of processors utilized for these computations is proportional to the number of chromosomes that are modified during each generation and because of the embarrassingly parallel implementation, optimization turn-around time is inversely proportional (approximately) to the number of chromosomes utilized. Thus, using an increased number of chromosomes improves turn-around efficiency without affecting MOGA convergence.

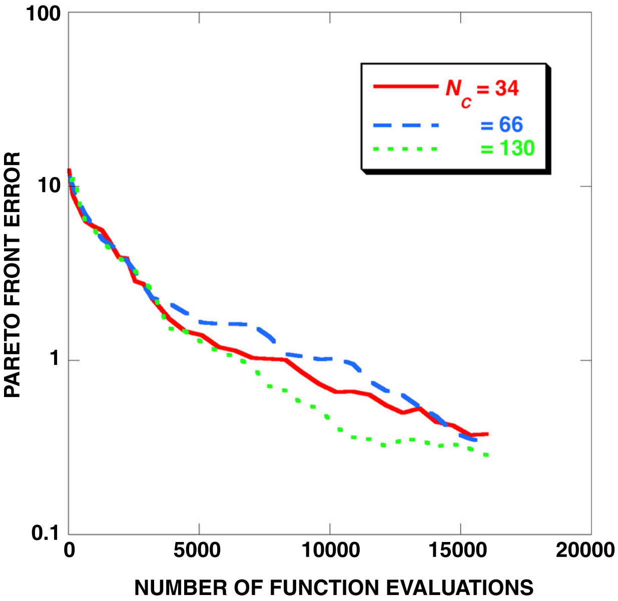


Fig. 9 Pareto front convergence histories for three population sizes each averaged over two solutions.

It is of interest to study the effect of wing placement on the overall wing-fuselage optimization problem. In the cases presented above, the vertical wing position, z_R, is constrained to lie between -0.15 and 0.15, while x_R and y_R are fixed at 1.5 and 0.0, respectively. As a result of the optimization, the value of z_R is at or near its upper constraint value, $z_R = 0.15$, over the entire computed Pareto front. In other words, a high-wing intersection is chosen by the optimization process regardless of position on the Pareto front (at least in the context of the present design space construction).

Figure 10 compares results for two different wing-fuselage optimization problems: a) the vertical wing position is utilized as one of the decision variables and thus is allowed to float $-0.15 \le z_R \le 0.15$ (this is the problem

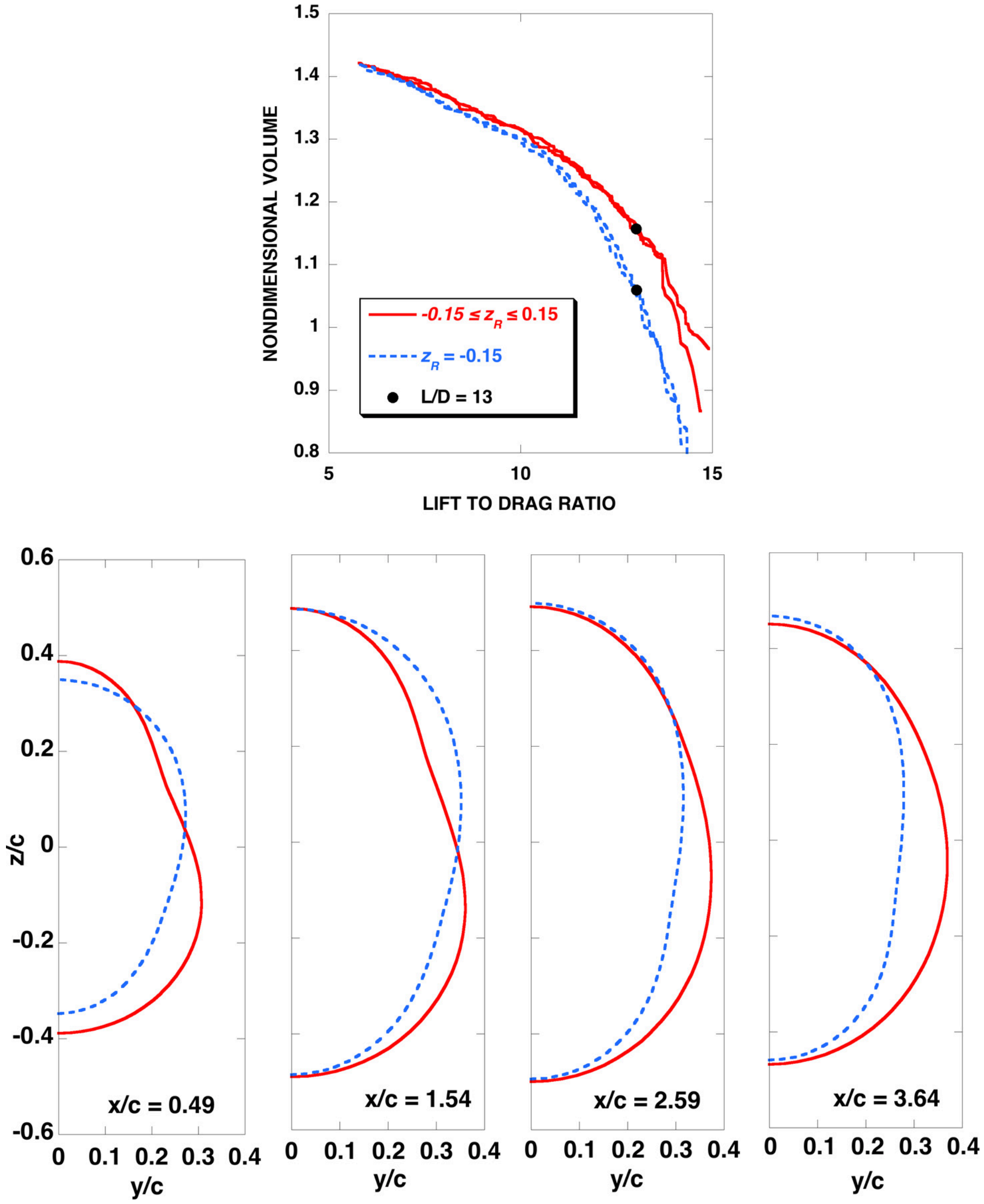


Fig. 10 Pareto front comparisons for two wing-fuselage optimization cases: a) floating wing position ($-0.15 \le z_R \le 0.15$), b) fixed wing position ($z_R = -0.15$). $C_L = 0.45$, $M_\infty = 0.84$. Selected fuselage cross-sections displayed at L/D = 13.

presented above) and b) the vertical wing position is constrained to be low mounted, that is, $z_R = -0.15$. The latter change is implemented by setting the maximum and minimum constraint values of z_R equal to -0.15 and by setting the corresponding z_R masking array value to zero. Thus, for this problem, $N_G = 42$, instead of the previous value of 43. Except for the vertical wing mounting position, all other aspects of these two problems (the two optimization objectives, the design space definition and the decision variable constraints) are the same. For this series of computations, $N_C = 66$, that is, 64 processors are used for each case. The flow conditions are $C_L = 0.45$ and $M_\infty = 0.84$.

The upper portion of Fig. 10 displays approximate Pareto fronts for the original wing-fuselage optimization problem as well as for the optimization that constrains the wing to be low-mounted. There are two curves plotted for each case, which are from solutions that are identical in every way except for the sequence of random numbers used in the various modification operators. Using different random number sequences for two or more cases that are otherwise identical, establishes an indicator for the statistical variation associated with the MOGA for that particular problem.

As can be seen, the low-mounted wing results are less efficient with respect to the minimum-drag Pareto front extreme. The low-mounted case achieves approximately the same level of drag, but only when vehicle volume is reduced. Note also that this effect is large in comparison to the statistical variation that exists between the two computations in each solution category. The effect of z_R on the maximum-volume Pareto front extreme is negligible.

Also displayed in the upper portion of Fig. 10 are two circular symbols showing a result from each Pareto front category that most closely matches a lift-to-drag ratio of 13. Selected fuselage cross sections taken from these two points are compared in the lower portion of Fig. 10. A total of four longitudinal stations are presented with $x/c = 0.49$, 1.54, 2.59 and 3.64. The solid curve corresponds to the original case in which the wing vertical mounting position was allowed to float, ending up with a high-mounted position. The dashed curve corresponds to the case in which the wing was constrained to be low mounted. Despite being at the same L/D location, the two area rulings are quite different—quite different at each longitudinal station. Close examination shows how the MOGA modified the area ruling to account for the shifted wing mounting position, taking volume from the upper fuselage for the high-mounted wing and from the lower fuselage for the low-mounted wing. It is also obvious from these cross-sectional profiles that the fuselage associated with the low-mounted wing contains a smaller amount of volume, which is consistent with the computed Pareto fronts displayed in the upper portion of Fig. 10.

VI. Conclusions

A multi-objective genetic algorithm (MOGA) optimization procedure has been evaluated. It uses real-number encoding to represent all design space decision variables as genes and populations of fixed size to go from generation to generation. Four modification operators are utilized to advance from one generation to the next. They include passthrough, random average crossover, perturbation mutation and global mutation. The standard output for this approach is an approximate Pareto front, which includes the best solutions from each objective, as well as a range of optimal "tradeoff" solutions in between.

The MOGA optimization procedure was utilized to determine a number of multi-objective optimal solutions for a transonic wing-fuselage problem in which the lift-to-drag ratio and the vehicle volume were maximized (all at fixed lift). In every case the MOGA optimization process was convergent.

The concept of aggressive mutation, which effectively biases the gene modification process toward gene values that reside on the design space boundary, was introduced. For the present problem, aggressive mutation produced a significant acceleration in MOGA convergence, primarily at the maximum-volume Pareto front extreme.

The present MOGA utilizes a masking array, which allows any gene or set of genes to be eliminated from the optimization process as a modifiable decision variable. This allows determination of the effect of a single gene or gene subset on the resultant Pareto front.

A limited parallelization efficiency study was performed involving the use of 32 to 128 processors. Population sizes ranging from 34 to 130 chromosomes were used. The present MOGA is embarrassingly parallel with regard to the number of chromosomes utilized in each generation, and the number of chromosomes did not significantly affect convergence efficiency of the MOGA scheme.

References

[1]Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA, 59–88, 1989.

[2]Davis, L., *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[3]Beasley, D., Bull, D. R., and Martin, R. R., "An Overview of Genetic Algorithms: Part 1, Fundamentals," *University Computing*, Vol. 15, No. 2, 1993, pp. 58–69.

[4]Beasley, D., Bull, D. R., and Martin, R. R., "An Overview of Genetic Algorithms: Part 2, Research Topics," *University Computing*, Vol. 15, No. 4, 1993, pp. 170–181.

[5]Deb, K., "Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems," *Evolutionary Computation*, Vol. 7, No. 3, 1999, pp. 205–230.

[6]Van Veldhuizen, D., and Lamont, G., "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art," *Evolutionary Computation*, Vol. 8, No. 2, 2000, pp. 125–147.

[7]Jimènez, J., Cuesta, P., and Abderramán, J., "Mixed Strategy in Genetic Algorithms: Domain's Reduction and Multirecombination," *European Congress on Computational Methods in Applied Sciences and Engineering*, Sept. 2000.

[8]Obayashi, S., and Tsukahara, T., "Comparison of Optimization Algorithms for Aerodynamic Shape Design," *AIAA Journal*, Vol. 35, 1997, pp. 1413–1415.

[9]Bock, K.-W., "Aerodynamic Design by Optimization," Paper 20, AGARD CP-463, 1990.

[10]Marco, N., Désidéri, J.-A., and Lanteri, S., "Multi-Objective Optimization in CFD by Genetic Algorithms," Institut National de Recherche en Informatique et en Automatique, Research Report No. 3686, April 1999.

[11]Naujoks, B., Willmes, L., Haase, W., Bäck, T., and Schütz, M., "Multi-Point Airfoil Optimization Using Evolution Strategies," *European Congress on Computational Methods in Applied Sciences and Engineering*, Sept. 2000.

[12]Quagliarella, D., and Della Cioppa, A., "Genetic Algorithms Applied to the Aerodynamic Design of Transonic Airfoils," AIAA Paper 94-1896-CP, 1994.

[13]Vicini, A., and Quagliarella, D., "Inverse and Direct Airfoil Design Using a Multiobjective Genetic Algorithm," *AIAA Journal*, Vol. 35, 1997, pp. 1499–1505.

[14]Hämäläinen, J., Mäkinen, A., Tarvainen, P., and Toivanen, J., "Evolutionary Shape Optimization in CFD with Industrial Applications," *European Congress on Computational Methods in Applied Sciences and Engineering*, Sept. 2000.

[15]Epstein, B., and Peigin, S., "A Robust Hybrid GA/ROM Approach to Multiobjective Constrained Optimization in Aerodynamics," 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper No. 2003–4092, June 2003.

[16]Anderson, M., Burkhalter, J., and Jenkins, R., "Missile Aerodynamic Shape Optimization Using Genetic Algorithms," *Journal of Spacecraft and Rockets*, Vol. 37, No. 5, Sept.–Oct. 2000, pp. 663–669.

[17]Anderson, M., and Gebert, G., "Using Pareto Genetic Algorithms for Preliminary Subsonic Wing Design," AIAA Paper No. 96-4023-CP, 1996.

[18]Sasaki, D, Obayashi, S., and Nakahashi, K., "Navier–Stokes Optimization of Supersonic Wings with Four Design Objectives Using Evolutionary Algorithm," AIAA Paper No. 2001–2531, 2001.

[19]Oyama, A., "Multidisciplinary Optimization of Transonic Wing Design Based on Evolutionary Algorithms Coupled with CFD Solver," *European Congress on Computational Methods in Applied Sciences and Engineering*, Sept. 2000.

[20]Ng, K. Y., Tan, C. M., Ray, T., and Tsai, H. M., "Single and Multiobjective Wing Planform and Airfoil Shape Optimization using a Swarm Algorithm," 41st Aerospace Sciences Meeting and Exhibit, AIAA Paper No. 2003–45, Jan. 2003.

[21]Obayashi, S., Yamaguchi, Y., and Nakamura, T., "Multiobjective Genetic Algorithm for Multidisciplinary Design of Transonic Wing Planform," *Journal of Aircraft*, Vol. 34, 1997, pp. 690–693.

[22]Oyama, A., and Liou, M.-S., "Multiobjective Optimization of Rocket Engine Pumps Using Evolutionary Algorithm," AIAA Paper No. 2001–2581, June 2001.

[23]Benini, E., "Three-Dimensional Multi-Objective Design Optimization of a Transonic Compressor Rotor," 16th AIAA Computational Fluid Dynamics Conference, AIAA Paper No. 2003–4090, June 2003.

[24]Oyama, A., and Liou, M., "Multiobjective Optimization of a Multi-Stage Compressor Using Evolutionary Algorithm," AIAA Paper No. 2002–3535, 2002.

[25]Holst, T. L., and Pulliam, T. H., "Evaluation of Genetic Algorithm Concepts using Model Problems, Part I: Single-Objective Optimization," NASA TM 2003-212814, Dec. 2003.

[26]Holst, T. L., and Pulliam, T. H., "Evaluation of Genetic Algorithm Concepts using Model Problems, Part II: Multi-Objective Optimization," NASA TM 2003-212813, Dec. 2003.

[27]Sobieczky, H., "Parametric Airfoils and Wings," *Recent Development of Aerodynamic Design Methodologies-Inverse Design and Optimization*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, Germany, 1999, pp. 72–74.

[28]Hicks, R., and Henne, P., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, 1978, pp. 407–412.

[29] Holst, T. L., "Multizone Chimera Algorithm for Solving the Full-Potential Equation," *Journal of Aircraft*, Vol. 35, No. 3, May–June 1998, pp. 412–421.

[30] Zitzler, E., Thiele, L. Laumanns, M., Fonseca, C. M., and da Fonseca, V. G., "Performance Assessment of Multiobjective Optimizers: An analysis and Review," *IEEE Transactions on Evolutionary Computation*, Vol. 7, No. 2, April 2003, pp. 117–132.

[31] Knowles, J. D., and Corne, D. W., "Approximating the Nondominated Front Using the Pareto Archived Evolution Stategy," *Evolution Computation,* Vol. 8, No. 2, 2000, pp. 149–172.

[32] Fonseca, C., and Fleming, P., "On the Performance Assessment and Comparison of Stochastic Multiobjective Optimizers," *Parallel Problem Solving From Nature IV,* edited by Voigt, H.-M., Ebeling, W., Rechenberg, I. and Schwefel, H.-P., Springer, Berlin, Germany, 1995, pp. 584–593.

[33] Holst, T. L., "Genetic Algorithms Applied to Multi-Objective Aerospace Shape Optimization," AIAA 1st Intelligent Systems Technical Conference, AIAA Paper No. 2004–6512, Sept. 2004.