JOURNAL OF AIRCRAFT
Vol. 48, No. 3, May–June 2011

Benefits of Incorporating Designer Preferences Within a Multi-Objective Airfoil Design Framework

Robert Carrese,* Hadi Winarto,† Jon Watmuff,‡ and Upali K. Wickramasinghe§
Royal Melbourne Institute of Technology, Bundoora, Victoria 3083, Australia

DOI: 10.2514/1.C001009

High-fidelity aerodynamic design problems are not easily managed, and identifying all Pareto-optimal design candidates is often unnecessary and computational exhaustive. We propose a variant of a multi-objective particle swarm optimization algorithm that draws on the domain knowledge of the designer to obtain solutions of interest. The swarm is guided by a reference point, which is viewed as an intuitive means of expressing the designer's preferred level of compromise that can ideally be based on some existing or target design. This hybrid methodology enhances the convergence proficiency and exploitation characteristics of the optimizer, due to the locally focused search effort. The algorithm attempts to identify a partial spread of Pareto-optimal designs that provide the most resemblance to the reference-point compromise. The algorithm is applied to a typical transonic airfoil design scenario in which the PARSEC parameterization model is used to represent candidate airfoil geometry, and a Reynolds-averaged Navier–Stokes solver is used to compute the aerodynamic coefficients. A hypervolume performance metric is applied to monitor convergence and solution spread. A data-mining technique using self-organizing maps is introduced and applied for postoptimization tradeoff analyses. Sample design problems are presented that depict the computational efficiency of the optimization framework and have the flexibility to adapt to disparate design philosophies.

I. Introduction

MODERN design methods draw on extensive computing facilities [1]. The significant progress achieved in automated optimization and accurate modeling procedures has paved the way for the development of innovative design strategies [2]. In the aerospace sector, this methodology is applied in the design of a range of aircraft systems and components and often encompasses multiple disciplines and design philosophies [1,3]. A field that has benefited considerably from automated design is airfoil aerodynamic shape optimization, which is an integral element of the aircraft multidisciplinary optimization framework. Deterministic and stochastic optimizers are coupled with variable-fidelity computational fluid dynamics solvers for elaborate design syntheses [4–9]. This involves applying an optimizer to identify the optimal perturbations to the airfoil geometry to adhere to given objective(s).

Shape optimization of transonic airfoils is employed to limit shock drag losses and reduce shock-induced boundary-layer instability at the design Mach number M and lift coefficient C_l. This often occurs at the expense of excessive pitching moments C_m, due to aft loading and performance degradation under offdesign conditions. To facilitate adequate performance over a multi-operational spectrum requires a search algorithm that is capable of handling multiple conflicting objectives, which follows the generic form [10],

$$\min_{\mathbf{x}\in S} f(\mathbf{x}) = \{f_1(\mathbf{x}), \dots, f_m(\mathbf{x})\} \qquad f_i\colon \mathbb{R}^n \to \mathbb{R} \tag{1}$$

$$c_j(\mathbf{x}) \geq 0, \quad \forall\ j = 1, \dots, p \tag{2}$$

where shape parameterization tools are used to construct the airfoil geometry mapped by the solution $\mathbf{x} = \{x_1, \dots, x_n\}$, within the n-dimensional design space $S \in \mathbb{R}^n$ bounded by $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ and any additional constraints c_j imposed on the design space. The m objectives f_i represent the drag C_d at a specific design condition, or perhaps the variation or fluctuation in drag over a design range. A tradeoff solution is considered Pareto-optimal, if, from that point in the design space, the value of any objective cannot be improved without deteriorating at least one other objective. To mathematically establish this concept, let two sets of decision vectors, $\mathbf{a}, \mathbf{b} \in S$. The decision vector $\mathbf{a}$ *strictly dominates* $\mathbf{b}$ (denoted by $\mathbf{a} \prec \mathbf{b}$) if

$$\forall\ i \in (1, \dots, m) f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \quad \text{and} \quad \exists\ i\colon f_i(\mathbf{a}) < f_i(\mathbf{b}) \tag{3}$$

It follows that a feasible solution $\mathbf{a}*$ is Pareto-optimal if there is no other feasible solution $\mathbf{a} \in S$ such that $f(\mathbf{a}) \prec f(\mathbf{a}*)$. The set of Pareto-optimal solutions is called the Pareto front. Gradient-based methods [11] or single-objective optimizers may be applied to identify the most *preferred* solution from the Pareto front, by aggregating the objectives into a single scalar using weights [12,13]. However, such approaches are sensitive to local minima present within the objective landscape and generally require that the objective function is continuous and differentiable. Flexibility to the designer is also limited, since multiple runs would be required to generate a set of tradeoff solutions; furthermore, the equivalent tradeoff solutions would not necessarily be representative of the actual Pareto-optimal solutions [14]. Alternatively, population-based evolutionary multi-objective (EMO) algorithms [10] are able to search for all Pareto-optimal solutions in one single run, providing the designer greater flexibility in selecting the most appropriate solution. Population-based heuristics have also gained much interest with the advent of cluster-based systems, since the computation of candidate groups is easily parallelized [1]. In this paper, an optimization algorithm based on the particle swarm optimization (PSO) heuristic is proposed. The PSO algorithm is a fairly recent addition to the existing list of EMO methods, first introduced by Kennedy and Eberhart [15]. We have selected PSO as it has proven to be an effective technique to facilitate single-objective optimization problems for its convergence proficiency in managing discontinuous and multimodal problems, as well as its quick and simple implementation. It has also since gained rapid popularity in the multi-objective optimization community [16].

Received 8 March 2010; revision received 1 January 2011; accepted for publication 17 January 2011.

*Ph.D. Candidate, School of Aerospace Mechanical and Manufacturing; robert.carrese@rmit.edu.au. Student Member AIAA.

†Associate Professor, School of Aerospace Mechanical and Manufacturing; hadi.winarto@rmit.edu.au. Senior Member AIAA.

‡Senior Lecturer, School of Aerospace Mechanical and Manufacturing; jon.watmuff@rmit.edu.au. Member AIAA.

§Research Assistant, School of Computer Science and Information Technology; uwickram@cs.rmit.edu.au.

Despite their reliability in global convergence, when confronted with a computationally challenging multi-objective design problem the performance of EMO methods may deteriorate, since they often rely on an excessive number of function evaluations [17]. They can become computationally intensive, since they are not guided by the differential landscape of the objective functions. The hybridization or fusion of different learning and adaptation techniques has the capacity to overcome these individual limitations [17]. For example, in aerodynamic design, researchers have integrated gradient-based methods into EMO methods, so-called *memetic* algorithms [6,18–20], to promote convergence and enhance exploitation. Focus has also shifted toward the integration of surrogate modeling, to replace the expensive solver [21–23]. In this paper, we propose a hybrid multi-objective framework, which relies on designer-preferences. Interactive methods incorporating designer-preferences originate from multicriteria decision making literature and require various levels of user input to obtain preferred solutions [24]. These classical methods are often disregarded, since the required user inputs (e.g., targets or bias weights) are generally not known in advance. There has, however, been increasing interest in coupling classical interactive methods to EMO algorithms as an intuitive way of specifying user preferences [25,26]. In recent studies user-preference methods have been applied to both genetic algorithms [27–29] and PSO [30]. All computing effort is thereby focused on preferred regions of the Pareto landscape to find only solutions of interest to the designer. This is advantageous for deceptive or multimodal landscapes or as the number of objectives increase, which is characteristic of aerodynamic design problems.

In this research we have drawn on the progress achieved in interactive optimization to develop a designer-driven airfoil design synthesis. We present the framework of our user-preference multi-objective particle swarm optimization (UP-MOPSO) algorithm. We then highlight the proficiency of this algorithm in managing transonic airfoil aerodynamic design problems of varying complexity and fidelity, over a conventional unguided search. A comparison mechanism using a hypervolume metric that measures convergence and solution spread is introduced. A user-preference-based design visualization module based on self-organizing maps (SOMs) is introduced, to facilitate the postoptimization tradeoff analysis. Conclusions are presented and avenues for future research are proposed.

II. User-Preference Optimization Framework

The process flow-cart of our proposed optimization framework is shown in Fig. 1. In transonic airfoil shape optimization, modifications to existing or new geometries are performed iteratively to conform to the best shape for a given objective or requirement. In a direct optimization procedure for the stipulated operating condition, deviations to the airfoil geometry will result in variations to the pressure and shear stress distributions that ultimately yield variations to the performance coefficients. The optimizer is used to determine the necessary perturbations to the airfoil geometry in order to minimize or maximize the desired performance coefficient(s) while adhering to any shape or performance constraints.

A. UP-MOPSO Algorithm

The UP-MOPSO algorithm applies the dominance criteria to explore the design space and identify the region of the global Pareto front. A reference point is projected on to the Pareto landscape by the designer to guide the search toward solutions of interest. This is an intuitive method to express the designer's preferred level of compromise, which could be based on some existing or target design. The multi-objective search effort is coordinated via a multi-objective PSO algorithm, which is based on the choreography of bird flock.

The PSO architecture is derived from the social-psychological tendency of individuals to learn from previous experience and emulate the success of others. Particles are represented by n-dimensional vectors $\mathbf{x}_i$ and $\mathbf{v}_i$, which are the particle position and velocity, respectively. Particles identify and exploit promising areas of S via coordinated movement, from the performance rating provided by the objective solver. There have been numerous modifications to the canonical particle swarm algorithm, which affect certain search characteristics [14]. It is advantageous to adopt a more diverse search tactic, which is more likely to overcome premature convergence so that particles do not commit prematurely to inferior optima [31]. We present a variant of the multi-objective particle swarm (MOPSO) heuristic, which incorporates user-defined preferences to direct all computing effort on preferred regions of the Pareto landscape. Implementing user preferences provides more diversity in the search, by ranking solutions according to their resemblance to the ideal compromise.

1. *Particle Flight*

In this research, the constriction type 1 framework proposed by Clerc and Kennedy [32] has been adopted. The ith particle of the swarm is accelerated toward its personal best position $\mathbf{p}_i$ and the global (or neighborhood) best position $\mathbf{p}_g$. The particle velocity magnitude is initialized randomly in the interval $[0, \mathbf{x}_{\max} - \mathbf{x}_{\min}]$. Half of the population's direction is reversed by setting the velocity to negative according to a coin toss. The updated position and velocity vectors at time $t+1$ are given by the following two equations:

$$\mathbf{v}_{i,(t+1)} = \chi[\mathbf{v}_{i,(t)} + \tilde{\mathbf{R}}_1[0,\varphi_1] \otimes (\mathbf{p}_{i,(t)} - \mathbf{x}_{i,(t)}) + \tilde{\mathbf{R}}_2[0,\varphi_2] \otimes (\mathbf{p}_{g,(t)} - \mathbf{x}_{i,(t)})] \tag{4}$$

$$\mathbf{x}_{i,(t+1)} = \mathbf{x}_{i,(t)} + \mathbf{v}_{i,(t+1)} \tag{5}$$

where $\tilde{\mathbf{R}}_1[0,\varphi_1]$ and $\tilde{\mathbf{R}}_2[0,\varphi_2]$ are two functions returning a vector of uniform random numbers in the range $[0,\varphi_1]$ and $[0,\varphi_2]$, respectively. The constants φ_1 and φ_2 are set to $\varphi/2$ where $\varphi = 4.1$. The constriction factor χ applies a dampening effect as to how far the particle explores within the search space, given as $\chi = 2/|2-\varphi-\sqrt{\varphi^2-4\varphi}|$.

2. *Integrating User Preferences*

Pareto-optimal solutions are all classified as equally optimal, hence there are several best solutions or swarm *leaders*. At each time steps, the best representative front found by the particles is stored within an archive [10,33]. Solutions in the archive are nondominated

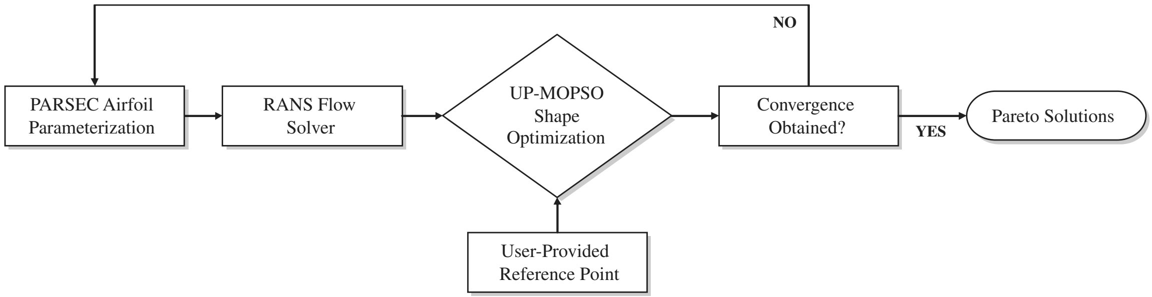

PARSEC Airfoil Parameterization
RANS Flow Solver
UP-MOPSO Shape Optimization
Convergence Obtained?
NO
YES
Pareto Solutions
User-Provided Reference Point

Fig. 1 User-preference airfoil shape optimization architecture.

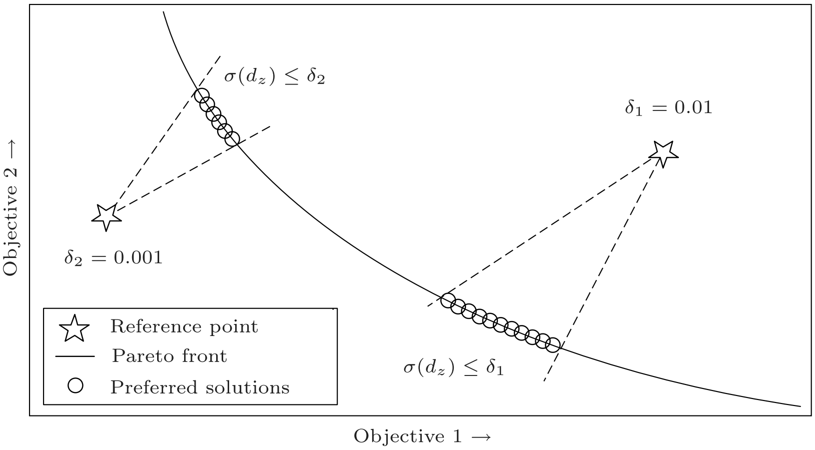


Fig. 2 User-preference particle swarm algorithm on a convex Pareto landscape.

and have no record of constraint violation. Each solution within the archive is identified as a possible leader. To provide additional guidance in selecting candidates for leadership from the archive, the reference-point method is adopted, following the work of Wickramasinghe and Li [30]. This module provides an intuitive criterion for selecting candidates for global leadership and assists the swarm to identify only solutions of interest to the designer. The guidance mechanism takes the form of a reference point $\bar{z}$, which is used to construct a distance metric to be minimized for $\mathbf{x} \in S$:

$$\text{minimize} \quad d_z = \max_{i=1:m}\{(f_i(\mathbf{x}) - \bar{z}_i)\} \tag{6}$$

where $\bar{z}_i$ is the ith component of the reference point or the aspiration value to the ith objective. The term w_i is the weight value for the ith fixed at one for the proceeding case studies. The designer generally has no prior knowledge of the topology and location of the Pareto front, therefore reference points may be ideally placed in any feasible or infeasible region, as shown in Fig. 2.

The reference point draws on the designer's experience to express a feasible compromise, rather than specific target values or goals. Similarly, the reference-point distance metric ranks or assesses a particle's success using one single scalar, instead of an array of objective values. Wickramasinghe and Li [34] later extended their approach to handle many-objective problems by replacing the dominance criteria entirely with the simpler and more efficient d_z distance metric. For our proposed algorithm, we apply the dominance criteria concurrently with Eq. (6) to find a feasible set of Pareto-optimal solutions with the most resemblance to $\bar{z}_i$. Nondominated solutions are first sorted based on d_z, of which the highest ranking solutions are selected as candidates for leadership and assigned randomly to each particle in the population. This will provide the necessary selection pressure for particles to move toward the preferred regions. The solution spread along the Pareto front is controlled by δ, as shown in Fig. 2. This parameter is defined as the maximum variance of the solutions' distance metric $\sigma(d_z)$. The extent of the solution spread is directly proportional to δ; evidently, as the value of δ increases, the influence of the reference-point location diminishes.

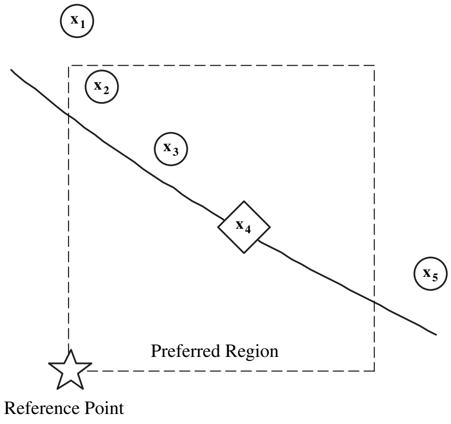


Fig. 3 Selection of leaders using reference-point method at time *t*.

3. *Leader Selection*

Swarm leaders are selected based on the position of nondominated solutions. After every population update, the top 10% ranked archive members are selected for leadership. Figure 3 shows a representative set of nondominated solutions on a two-objective landscape approaching the global Pareto front at time t. Since the solutions $\mathbf{x}_1$ and $\mathbf{x}_5$ lie outside the preferred region (i.e., specified by δ), candidates for leaders are selected using Eq. (6). If the repository limit N_b is breached, solutions with the highest reference-point distances have a high selection pressure for removal. To avoid particle clustering a crowding distance metric is applied [33,35,36]. The crowding distance of a particle is an estimate to the density of solutions in its immediate surroundings. It calculates the average Euclidean distance of the solution with respect to all objectives. If all nondominated solutions exist within the preferred region, global leaders are obtained from candidates with the highest crowding distance values. This ensures that particles within the swarm are guided to explore sparsely populated areas. If the archive limit N_b is breached, solutions with the lowest crowding distances have a high selection pressure for removal. Particles in densely populated regions are therefore more likely to be replaced by new nondominated solutions.

4. *Handling Constraints*

When comparing particles for admission in the archive, a constraint-dominance procedure is applied following the work of Deb et al. [36]. For each particle, $c_j = (c_1, \ldots, c_p)$, where p is the number of constraints and $c_j \geq 0$ is the violation of the constraint. A solution **a** constraint dominates **b** if any of the following criteria are met:

1) Solution **a** is feasible and solution **b** is not.
2) Both solutions **a** and **b** are infeasible but $c(\mathbf{a}) < c(\mathbf{b})$.
3) Both solutions **a** and **b** are feasible but $d_z(\mathbf{a}) < d_z(\mathbf{b})$.

Therefore, if both solutions are deemed to satisfy the constraint values, the more *preferred* particle is admitted for entry. However, if both particles are infeasible, the particle with the overall least constraint violation is considered the better solution.

5. *Stagnant Particle Mutation*

Despite the additional guidance provided by user preferences and the diversity inherent within the proposed topology, the search proficiency of the swarm may deteriorate when confronted with a highly multimodal problem [16]. It follows that as the velocity of the particles approach zero (i.e., $\mathbf{v} \to 0$), the swarm is unable to generate new leaders. This could lead to premature convergence as a result of the swarm being trapped within a local front. Kennedy and Eberhart [15] observed that the probability of such an occurrence may be reduced by introducing some element of flight turbulence or craziness. Turbulence, or *mutation*, issues a random variation in the particle's flight trajectory or position. The mutation operator (when applied appropriately) is very effective at generating new leaders. If mutation is incorrectly implemented, it becomes destructive and deteriorates the natural explorative capabilities of the swarm.

A Gaussian mutation operator is applied to particles within the archive. To ensure that the mutation is nondestructive, only mutated particles that provide a lower d_z are permitted for entry. The mutation operator is triggered when archive solutions have become stagnant for a number of successive time steps. The percentage of mutated particles p_{mut}, steadily reduces as the repository reaches maximum capacity to avoid unnecessary evaluations. When a particle is selected for mutation, there is a 10% probability that a dimension will be mutated. For a given parent i and dimension j, Gaussian mutation is applied as follows:

$$x_{i,j} = x_{i,j} + N(\mu, \sigma^2) \tag{7}$$

where $N(\mu, \sigma^2)$ represents a random number of mean $\mu = 0$ generated from the normal distribution curve. The standard deviation, σ^2 is initially given a value of 0.1 and steadily reduces to 0.05 as the algorithm advances to promote exploitation.

6. *UP-MOPSO Pseudocode*

The UP-MOPSO algorithm is summarized in the following steps. The stopping criterion for the algorithm is based on the maximum number of function evaluations as specified by the user.

```
1) Specify reference point $\bar{z}$ and solution spread $\delta$.
2) Initialize the swarm population:
    a) The position $\mathbf{x}_i$, velocity $\mathbf{v}_i$, and personal best position $\mathbf{p}_i = \mathbf{x}_i$.
    b) Evaluate each particle in the population; time steps $t := 0$.
3) Time steps $t := t + 1$.
4) Apply domination criteria to update archive, $Q$:
    a) Calculate the distance metric $d_z$ [via Eq. (6)] of each swarm particle.
    b) Identify nondominated solutions and include in $Q$; if $\sigma(d_z) \leq \delta$, then rank $Q$ by min $d_z$, else rank $Q$ by crowding distance.
    c) If $Q$ limit is breached, if $\sigma(d_z) \leq \delta$, then rank $Q$ by min $d_z$, else rank $Q$ by crowding distance; lowest-ranked members are removed.
5) Update swarm population:
    a) Randomly assign highest-ranked members of $Q$ to particles.
    b) Update particle position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$ as per Eqs. (4) and (5); evaluate each particle in the population.
6) Apply mutation operator if consistent improvement in $Q$ is not recorded.
7) Update $\mathbf{p}_i$ if it dominates existing $\mathbf{p}_i$.
8) If not maximum number of evaluations reached, then goto 3.
```

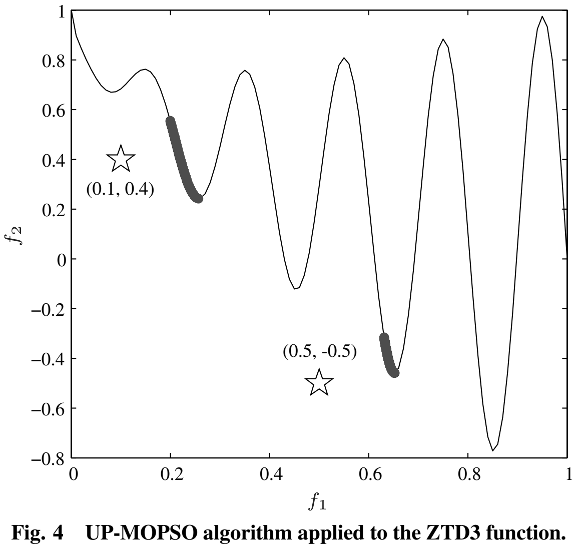


Fig. 4 UP-MOPSO algorithm applied to the ZTD3 function.

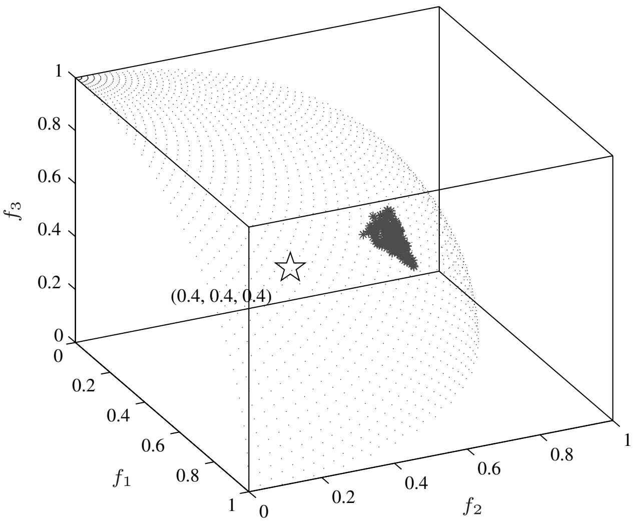


Fig. 5 UP-MOPSO algorithm applied to the DTLZ2 function.

Figures 4 and 5 demonstrate the UP-MOPSO algorithm applied to benchmark mathematical test functions. The ZTD3 [37] test function has a discontinuous Pareto landscape, which is representative of aerodynamic design problems. The DTLZ2 [38] test function is multimodal and has a concave Pareto surface. A population of 100 particles is flown for 100 time steps, with $\delta = 5 \times 10^{-2}$. It is clearly observed that only particles that are closest to the preferred regions reside on the Pareto fronts.

B. Flow Solver

In this study, the general-purpose finite volume code FLUENT [39] is used. A steady-state, fully coupled, pressure-based numerical procedure is adopted with third-order discretization. The momentum equations and pressure-based continuity equation are solved concurrently, with the Courant–Friedrichs–Lewy number set at 200. The one-equation Spalart–Allmaras [40] turbulence model is adopted with the assumption that turbulent flow acts over the entire airfoil.

GAMBIT is used to generate a parabolic C-type grid (see Fig. 6) stretching 25 chord lengths aft and normal of the airfoil section. Resolution of the C-grid is 460 × 65 providing an affordable mesh size of approximately 30,000 elements. The first grid point is located 2.5×10^{-4} units normal to the airfoil surface, resulting in an average y-plus value of 120. In the interest of robust and efficient convergence rates, a full-multigrid initialization scheme is employed, with coarsening of the grid to 30 cells. In the full-multigrid initialization process, the Euler equations are solved using a first-order discretization to obtain a flowfield approximation before submitting to the full iterative calculation. FLUENT simulations are performed in parallel through the MATLAB Distributed Computing Server [41].

C. PARSEC Shape Parameterization

Airfoil geometry has been represented using a variety of methods. In shape optimization, it is essential that a parameterization model be concise and robust, yet flexible to increase breadth in the range of shapes explored. Most importantly, however, airfoil parameterization is an important contributing factor to the efficiency of the design framework, since it will define the objective landscape and the topology of the design space [42]. Coordinates have been directly used to provide the contour of the airfoil geometry through interpolation methods such as B-splines [43], Bézier curves [44], and nonuniform rational B-splines [45]. Despite the flexibility offered by local parameterization methods, the large number of variables may

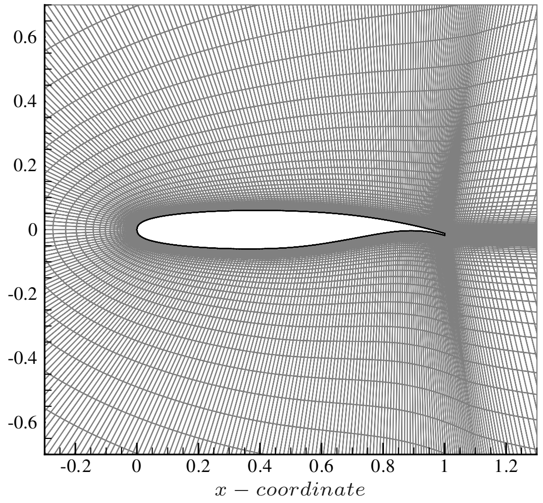


Fig. 6 RANS grid for transonic optimization case studies.

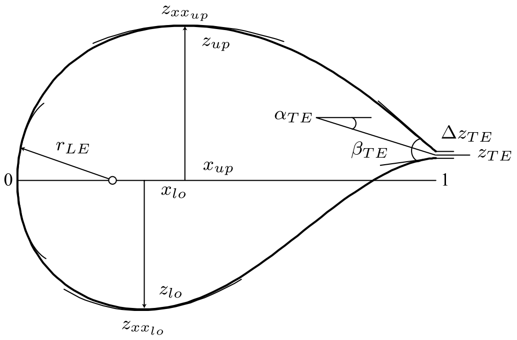


Fig. 7 Original PARSEC method.

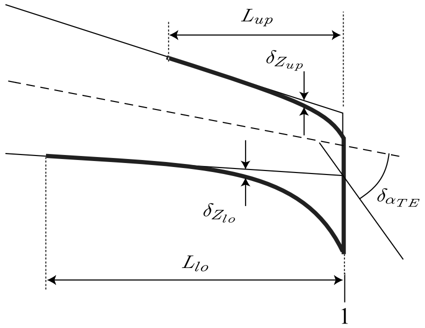


Fig. 8 Modified PARSEC method with additional trailing-edge control.

prove to deteriorate the convergence rate of global design models. The development of efficient parameterization models has therefore been given significant attention, to increase the flexibility of geometrical control with a minimum number of design variables.

A common method for airfoil shape parameterization is the PARSEC method [46]. It has the advantage of strict control over important aerodynamic features, and it allows independent control over the airfoil geometry for imposing shape constraints. In this study, the modified PARSEC method is adopted from Jahangirian and Shahrokhi [47]. The modification includes an additional variable to provide further control over the trailing-edge curvature. Additional control at the trailing edge is beneficial in order to reduce the probability of downstream boundary-layer separation, giving rise to increased drag values.

Illustrated in Fig. 7 are the basic 11 parameters that are used to completely define the profile geometry. The description and range of each variable is shown in Table 1. The trailing-edge thickness variable $\Delta z_{\text{TE}} = 0$ has been removed due to meshing constraints. Furthermore, the trailing-edge coordinate $z_{\text{TE}} = 0$ has been removed, since the control that this variable provides is inherent within the adopted modification, as shown in Fig. 8. The modification to the trailing edge is applied to the upper and lower surfaces as follows:

$$\delta_Z = \frac{L \cdot \tan \delta_{\alpha_{\text{TE}}}}{2\mu\tau}[1 + \eta \cdot x^\tau - (1 - x^\tau)^\mu] \tag{8}$$

where the constants η, μ, and τ are set to 0.8, 2, and 6, respectively. The modification is applied over the entire surface, such that $L_{\text{up}} = L_{\text{lo}} = 1$.

Given the designer's refined knowledge about the occurring flow phenomena, the PARSEC parameters may be restricted to conform to a specific family of airfoils [6]. This is achieved through inverse mapping of benchmark profiles that have been developed (either by experimental or computational methods) to perform favorably in transonic flow. Parameter ranges were determined (see Table 1) from a statistical sample of the benchmark profiles. Defining the airfoil boundaries through inverse mapping as opposed to arbitrarily selecting boundaries is advantageous to bypass poorly performing areas of the design space. A thickness constraint is explicitly defined through the parameter ranges as approximately 9.75% of chord.

III. Experimental Validation

The computational efficiency of the UP-MOPSO algorithm is highlighted through a series of computational experiments. The experiments examine the ability of the framework to handle multiple objectives and constraints and disparate operational requirements.

A. Hypervolume Metric

There are several popular performance metrics to monitor solution spread and accuracy for multi-objective optimization [37]. However, these metrics are only suitable for problems where the global Pareto front is known a priori. For engineering design problems, and in particular airfoil shape optimization problems, the global Pareto front is generally unknown. Hence, a feasible performance metric is required that does not rely on this information to compare various algorithms. The hypervolume (HV) metric is one such method [48]. The HV metric provides a single measurement to assess both the convergence and spread of solutions. For this application, the

Table 1 PARSEC parameter ranges for transonic optimization

Designation	Description	Variable	Lower Bound	Upper Bound
1	Leading-edge radius	r_{LE}	0.0063	0.0151
2	Trailing-edge direction	α_{TE}	0.2405(-)	0.0026(-)
3	Trailing-edge wedge angle	β_{TE}	0.0655	0.2618
4	Upper-crest abscissa	x_{up}	0.3170	0.5250
5	Upper-crest ordinate	z_{up}	0.0497	0.0683
6	Upper-crest curvature	$z_{xx_{\text{up}}}$	0.5135(-)	0.2393(-)
7	Lower-crest abscissa	x_{lo}	0.2835	0.3418
8	Lower-crest ordinate	z_{lo}	0.0603(-)	0.0478(-)
9	Lower-crest curvature	$z_{xx_{\text{lo}}}$	0.2535	0.8405
10	Trailing-edge curvature	$\delta_{\alpha_{\text{TE}}}$	0.0080(-)	0.3696

standard HV calculation procedure must be modified to include only the solutions points within the preferred regions, not the entire Pareto front.

The HV metric calculates the total volume bounded by the solutions on the Pareto front and a selected point in the objective space. The selected point is termed the *nadir point*. At the nadir point, all objectives are at their worst values simultaneously. The nadir point $\mathbf{x}_{\text{nad}}$ is calculated as

$$\mathbf{x}_{\text{nad}} = [f_1^{\text{nad}}(\mathbf{x}), \ldots, f_M^{\text{nad}}(\mathbf{x})] \tag{9}$$

where

$$f_i^{\text{nad}}(\mathbf{x}) = \max_{j=1,\ldots,N_s} \{f_i(\mathbf{x}_j)\}$$

The formal definition of HV is the Lebesgue measure Λ of the union of all hypercubes a_i defined by a nondominated point $\mathbf{b}_i$ and the nadir point $\mathbf{x}_{\text{nad}}$. The corresponding HV value is the sum of all these volumes, as denoted in Fig. 9:

$$\text{HV}\,(B) = \Lambda\left(\left\{\bigcup_i a_i | \mathbf{b}_i \in B\right\}\right) = \Lambda\left(\bigcup_{\mathbf{b}_i \in B} \{\mathbf{x} | \mathbf{b}_i \prec \mathbf{x} \prec \mathbf{x}_{\text{nad}}\}\right) \tag{10}$$

When conducting experimental validation with stochastic algorithms, it is only justifiable that multiple runs are conducted, in order to eliminate the influence of random occurrences. The nadir point is calculated from the final combined population and is subsequently used to obtain the HV curve for each independent simulation. When comparing different algorithms, that which provides the greatest HV value is considered to be superior, since it gives a measure of both the spread and the closeness of the solution to the Pareto front. For interactive optimization, the objective is to locate nondominated solutions in preferred regions. The HV metric must therefore be modified to account for such algorithms.

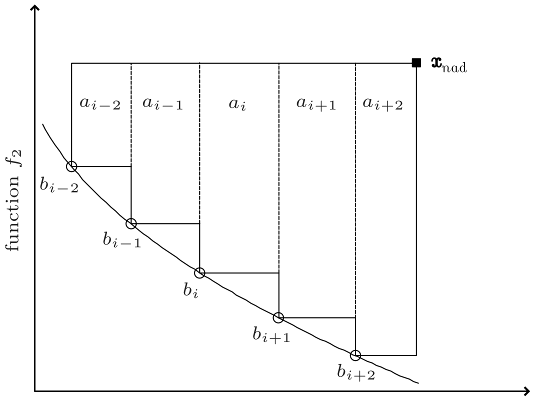


Fig. 9 Definition of hypercubes in a two-objective Pareto landscape.

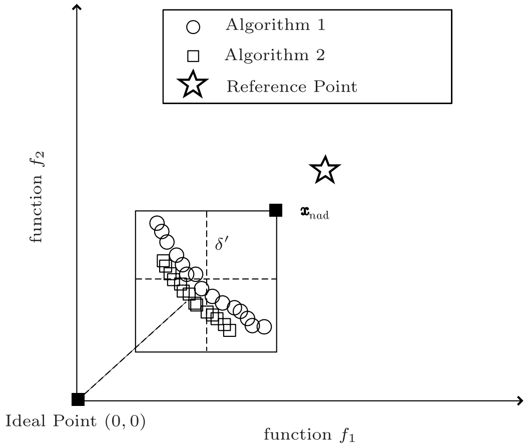


Fig. 10 Volume bounded by solution closest to ideal point.

The Euclidean distance is calculated for each solution point in this population from the ideal point. The solution point with the lowest Euclidean distance is selected to define a volume. For a two-objective minimization problem, the ideal point is positioned at $(0, 0)$. A volume is then defined around the solution with the lowest Euclidean distance, as demonstrated in Fig. 10. The value δ' is prescribed such that a volume is defined having $2\delta'$ for each objective. Solutions that reside outside the preferred volume are simply excluded from the calculation. The remaining solutions are used to obtain the nadir point, which will be located within the defined volume. The calculation of the HV values for each independent run is then performed using this common nadir point.

B. Experiments

Two computational experiments have been devised to compare the UP-MOPSO algorithm with and without the reference-point method. Through these experiments, we are attempting to highlight the benefits of incorporating the preferences of the designer within a typical design optimization framework. We hereby refer to the UP-MOPSO algorithm without the reference-point method as the unguided algorithm. The flow solver adopted in the subsequent experiments is a low-fidelity Euler solver. A C-grid resolution of 196×46 results in a cell count of approximately 9000. The Euler solver is not computationally intensive and requires only a few seconds per function evaluation, yet still captures important shock-flow phenomena. The design space is bounded according to Table 1. For all simulations, a swarm population of $N_s = 100$ is initialized and flown for $t = 200$ time steps. All objectives are to be minimized. For each problem, two independent simulations are conducted, each for the UP-MOPSO algorithm and the unguided algorithm. The final HV calculation is a resultant of the average progress of each algorithm for all independent simulations.

1. Multipoint Design

This experiment is illustrative of a multipoint design philosophy. The objectives are to minimize $f_1 = C_d$ for $M = 0.75$ and $C_l \geq 0.70$ and to minimize $f_2 = C_d$ for $M = 0.79$ and $C_l \geq 0.40$. The lift constraint is not satisfied explicitly, due to the computational expense involved to vary the incidence angle α within the solver environment. Alternatively, there are an additional two decision variables for this problem, which are the angles α_{f_1} and $\alpha_{f_2} \in [-1°, 1°]$. An additional constraint on the moment coefficient is imposed as $C_m^2 \leq 0.04$ for both operating conditions. The reference point is provided as $\bar{z} = (0.0062, 0.0033)$, suggestively placing more emphasis on the second objective. A solution spread of $\delta = 5 \times 10^{-3}$ is requested.

The average computational time required per time steps on an Intel Core2 Duo 2.66 GHz CPU is approximately 40 min. The reference point $\bar{z}$ is observed to be situated in a dominated region of the Pareto landscape. For this example, both algorithms have arrived to the region of the global Pareto front, as shown in Fig. 11. However, as dictated by Fig. 12, the UP-MOPSO algorithm is observed to have a far superior convergence rate and provides a considerably more precise set of solutions. The region where the HV curve is observed to have a high gradient value suggests that the swarm is in the exploration phase. As the algorithm progresses, the small increments in the HV value arise from the exploitation of individual solutions and solution uniformity. It follows from Fig. 12 that the UP-MOPSO algorithm arrives to the global Pareto front in 20% of the allocated time steps as opposed to 70% for the unguided algorithm for this problem.

The location of $\bar{z}$ has resulted in more priority being placed on the second design condition. If the aspiration value of the second design condition were to be relaxed, the UP-MOPSO algorithm would have found airfoils that exhibit greater aerodynamic improvement at the $M = 0.75$ condition. The reference point can therefore be viewed as a means of expressing a feasible compromise, rather than target values. Since the feasibility of the location of the reference point is not of

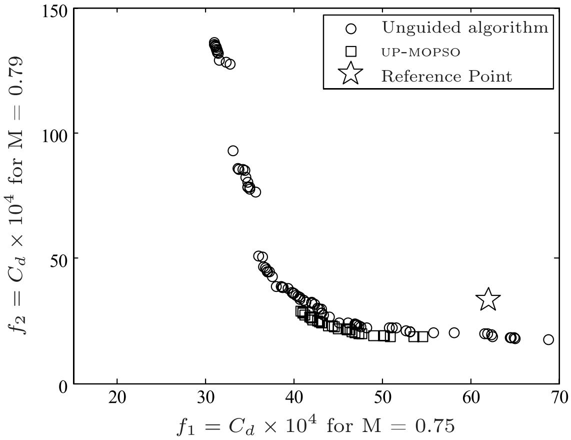


Fig. 11 Final Pareto solutions obtained for multipoint design.

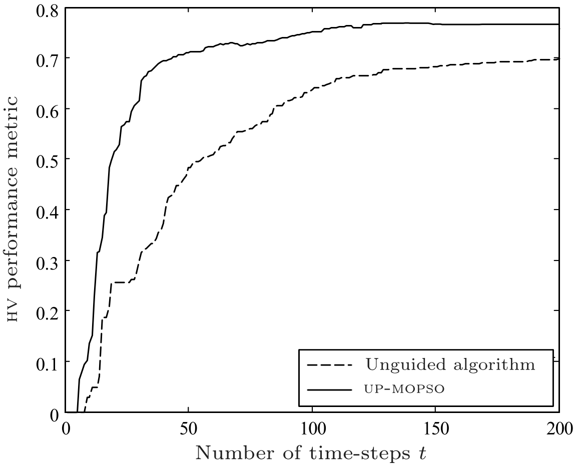


Fig. 12 HV performance metrics for multipoint design.

concern, any suitable compromise may be stipulated at the designer's discretion.

2. Robust Single-Point Design

This problem adopts a fairly simplistic single-point design philosophy. The objectives are provided as $f_1 = C_d/C_l^2$ and $f_2 = C_m^2$ for the design Mach number of $M = 0.80$. Additionally, the algorithm is required to find solutions that do not exhibit a high drag rise with an abrupt change in velocity, where $f_3 = \partial C_d/\partial(10 \cdot M)$ for $\Delta M = 0.03$. The angle α is varied within the range $[-1°, 1°]$. The reference point is given as $\bar{z} = (0.025, 0.014, 0.012)$ with a solution spread of $\delta = 1 \times 10^{-2}$.

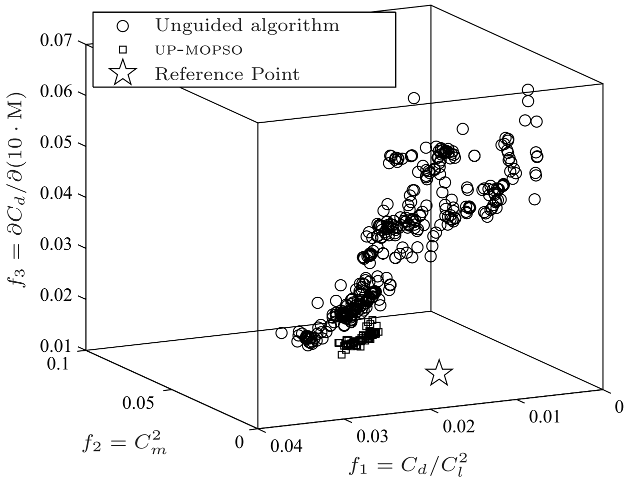


Fig. 13 HV performance metrics for robust single-point design.

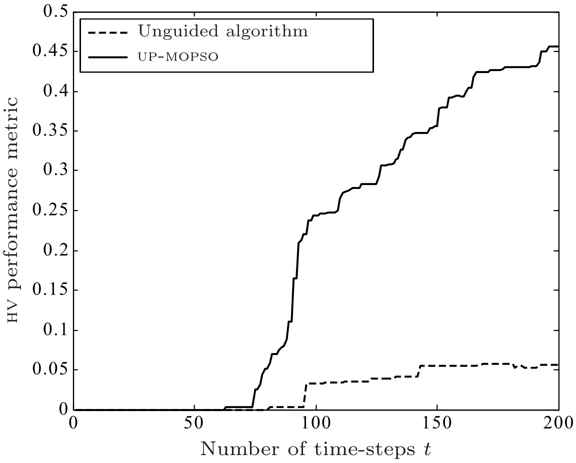


Fig. 14 Final Pareto fronts for robust single-point design.

The average computational time required per time steps on an Intel Core2 Duo 2.66 GHz CPU is approximately 40 min. It is observed from Fig. 13 that the benefits of adopting the UP-MOPSO algorithm increase as the number of objectives are increased. The unguided algorithm is shown to explore all regions of the Pareto landscape and, as such, has resulted in premature convergence to a suboptimal front. On the contrary, the UP-MOPSO was observed to arrive at the global Pareto front (despite the infeasible location of $\bar{z}$) within the allocated number of time steps and with sufficient time to exploit the individual solutions found. Since the HV metric shown in Fig. 14 is a measure of both convergence and solution spread, the increments in the HV value as the algorithm progresses are accredited to improvements in solution uniformity and spread.

IV. Design Problems

From the experiments conducted in the previous section, the UP-MOPSO framework has shown to be adaptable to various design philosophies, and the design space is sufficient to explore a wide range of operating conditions. The interactive UP-MOPSO algorithm provides a more proficient convergence rate and is also observed to have greater exploitation characteristics over the unguided algorithm. The following design cases are attempted using the UP-MOPSO algorithm and the high-fidelity Reynolds-averaged Navier–Stokes (RANS) solver.

A. Robust Single-Point Design

This problem is representative of the experiment conducted in Sec. III.B.2. The algorithm is applied to the redesign of the

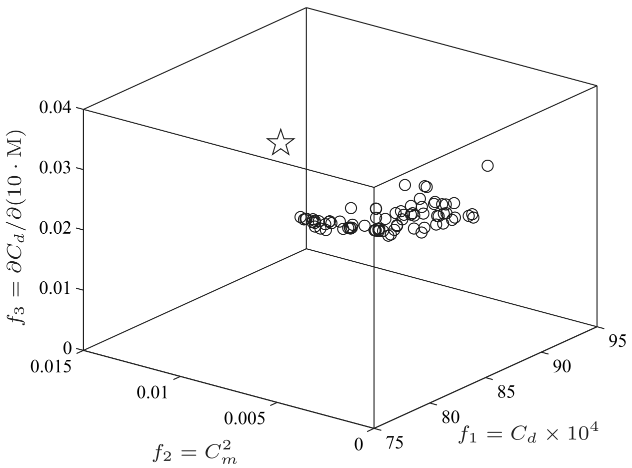


Fig. 15 Final Pareto front for robust single-point transonic design.

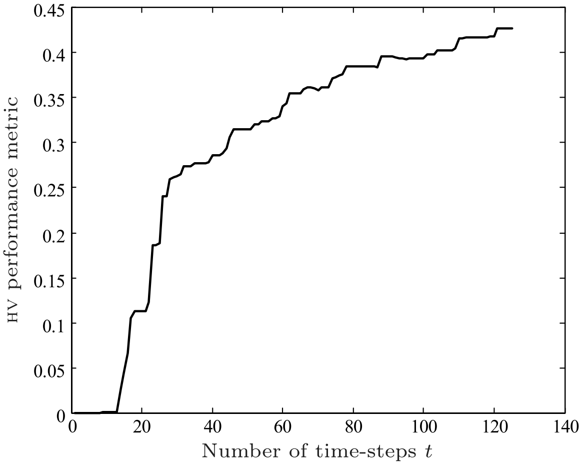


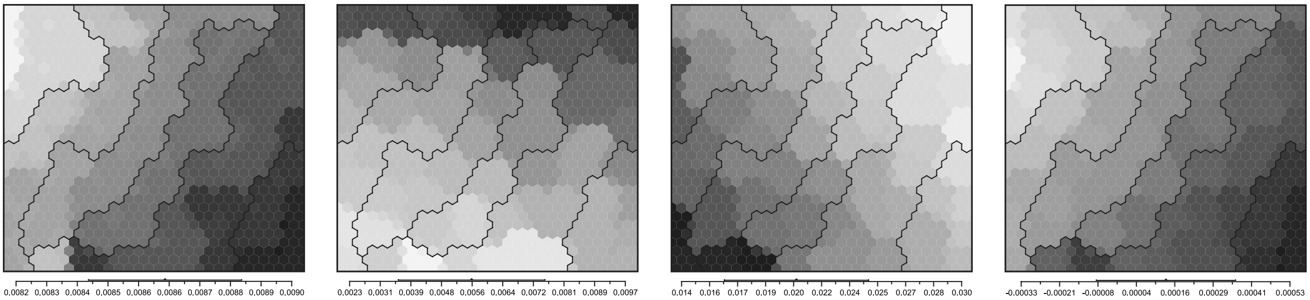
Fig. 16 HV performance curve for robust single-point transonic design.

NASA-SC(2)0410 reference airfoil for an abrupt change in flow velocity. The objectives are stipulated as $f_1 = C_d \times 10^4$ and $f_2 = C_m^2$ for $M = 0.79$, $C_l \geq 0.4$, and $f_3 = \partial C_d / \partial(10 \cdot M)$ for an abrupt change in ΔM of 0.03. The angle α is varied within the range $[-1°, 1°]$. The aspiration values are obtained using the RANS solver, given as $\bar{z} = (85.1, 0.0106, 0.0297)$.

A swarm population of $N_s = 100$ is flown in parallel for $t = 200$ time steps. A limit is placed on the repository as $N_b = 75$. For any t, the number of particles selected for mutation is restricted to 30. The simulation is performed with 10 cores using the Tango cluster of the Victorian Partnership for Advanced Computing (VPAC). Each compute node has two AMD Barcelona 2.3 GHz quad core processors. The average computational time required per time steps is approximately 1.25 h. The optimization sequence was terminated after $t = 125$ time steps. It is observed from Figs. 15 and 16 that the algorithm begins to enter and populate the preferred region of the global Pareto front by $t = 35$, beyond which the small increments in the HV value are a resultant of the algorithm improving solution uniformity and precision. The algorithm was terminated at this stage, since no improvement in the preferred solutions was observed.

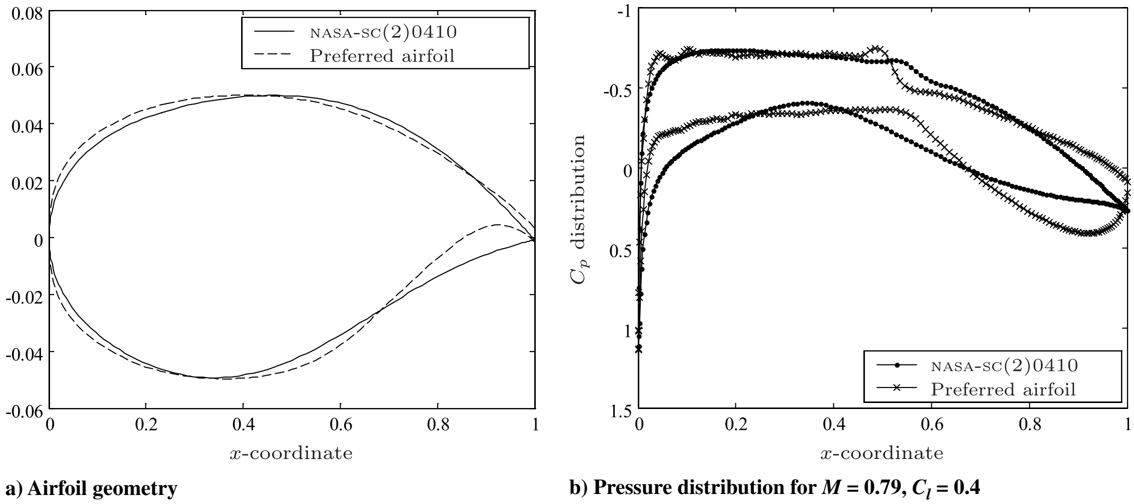
The concept of the user-preference module also provides a feasible means of selecting the most appropriate solutions [21]. For example, solutions may be ranked according to how well they represent the compromise provided by $\bar{z}$. The reference-point distance also provides a feasible means of selecting the most appropriate solutions. For example, solutions may be ranked according to how well they represent the reference-point compromise. To illustrate this concept, we use a multidimensional data-mining technique. In this study, we use SOMs to visualize the interaction of the objectives with the reference-point compromise. Clustering SOM techniques are based on a technique of unsupervised artificial neural network [49] that can classify, organize and visualize large sets of data from a high-to-low dimensional space [6,50]. A neuron is associated with a weighted vector of m inputs. Each neuron is connected to its adjacent neurons by a neighborhood relation and forms a two-dimensional hexagonal topology. The learning algorithm within SOM will attempt to increase the correlation between neighboring neurons to provide a global representation of all solutions and their corresponding resemblance to the reference-point compromise [50].

The SOM analyses have been conducted using Viscovery SOMine 5.2. The axes have no meaning, and the two-dimensionality of the charts is to facilitate a qualitative visual analysis. Figure 17 features the postoptimization SOM tradeoff study, organized by six SOM-ward clusters. Solutions that yield negative d_z values indicate success in improvement over each aspiration value. Solutions with positive



a) SOM colored by f_1 **b) SOM colored by f_2** **c) SOM colored by f_3** **d) SOM colored by d_z**

Fig. 17 SOM charts to visualize optimal tradeoffs between objectives (lighter shading denotes value minima).



a) Airfoil geometry **b) Pressure distribution for $M = 0.79$, $C_l = 0.4$**

Fig. 18 Comparison between preferred solution and reference NASA-SC(2)0410 airfoil.

Table 2 Preferred airfoil results for robust single-point transonic design

Airfoil	M	C_l	$C_d \times 10^4$	C_m	$\partial C_d/\partial(10 \cdot M)$
Preferred	0.79	0.40	81.7	−0.0713	0.0264 at $\Delta M = 0.03$
NASA-SC(2) 0410	0.79	0.40	85.1	−0.103	0.0297 at $\Delta M = 0.03$

d_z values do not surpass each aspiration value, but provide significant improvement in at least one other objective. Each of the node values represent one possible Pareto-optimal solution that the designer may select. The SOM chart colored by d_z is a measure of how far a solution deviates from the preferred compromise. However, the concept of the preferred region ensures that only solutions that slightly deviate from the compromise dictated by $\bar{z}$ are identified. Following the SOM charts, it is possible to visualize the preferred compromise between the design objectives that is obtained. The chart of d_z closely follows the f_1 chart, which suggests that this objective has the highest priority. If the designer were inclined toward another specific design objective, then perhaps solutions that place more emphasis on the other objectives should be considered.

For this problem, the *preferred* airfoil is selected to be that with the lowest d_z. The preferred airfoil geometry is shown in Fig. 18a in comparison with the NASA-SC(2)0410. The preferred airfoil has a thickness of 9.81% chord and maintains a moderate curvature over the entire upper surface. There is relatively no aft curvature on the lower surface and trailing edge, which results in a reduced C_m value. For the specified design condition, the airfoil requires an angle $\alpha = 0.812°$ to satisfy the lift constraint. Values for the objective functions are displayed in Table 2. The preferred airfoil has a significantly weaker shock in comparison with the NASA-SC(2) 0410 reference airfoil, highlighted by the C_p distributions of Fig. 18b and the static pressure contours of Fig. 19b.

An improvement of 4.1, 30.7, and 11.1% is recorded over the respective aspiration values. Referring to Fig. 18b, the reduction in C_d is attributed to the significantly weaker shock that appears slightly downstream of the supercritical shock position. This is predominately due to the upper-surface curvature, which does not produce excessive acceleration. The improvement in C_m is clearly visible from the reduction in aft loading. Along with the improvement at the required design condition, the preferred airfoil exhibits a lower C_d rise by comparison. The objective considers an abrupt change in Mach number, which implies that C_l is not corrected. Figure 20 shows the C_d distribution with Mach number for the preferred airfoil compared with the reference airfoil. Also demonstrated is the solution that provides the most robust design. The most robust design is clearly not obtained at the expense of poor performance at the design condition, due to the compromising influence of $\bar{z}$.

B. Multipoint Design

The multipoint problem established in Sec. III.B.1 is extended to incorporate a third design point. The objectives are to minimize $f_1 = C_d$ for $M = 0.75$ and$C_l \geq 0.70$, $f_2 = C_d$ for $M = 0.78$ and $C_l \geq 0.60$, and $f_3 = C_d$ for $M = 0.84$ and $C_l \geq 0.40$. A constraint on the moment coefficient is imposed as $C_m^2 \leq 0.04$ for each operating condition. In this case, there are an additional three decision variables for this problem, which are the angles α_{f_1}, α_{f_2}, and $\alpha_{f_3} \in [-1°, 1°]$. The aspiration values are obtained from NASA-SC (2) airfoil RANS results, as shown in Table 3. A solution spread of $\delta = 5 \times 10^{-3}$ is requested.

Table 3 NASA-SC(2) RANS results used as aspiration values for multipoint design

Airfoil	M	C_l	$C_d \times 10^4$
NASA-SC(2) 0712	0.75	0.70	117.9
NASA-SC(2) 0610	0.78	0.60	104.7
NASA-SC(2) 0410	0.84	0.40	235.1

A swarm population of $N_s = 100$ is flown in parallel for a maximum number of $t_{\max} = 200$ time steps. The simulation is again performed with 10 cores using the VPAC Tango cluster, yielding an average computational time per time steps of approximately 1.75 h. The optimization sequence was terminated after $t = 150$ time steps, since no improvement in the 75 obtained Pareto-optimal solutions (shown in Fig. 21) was recorded.

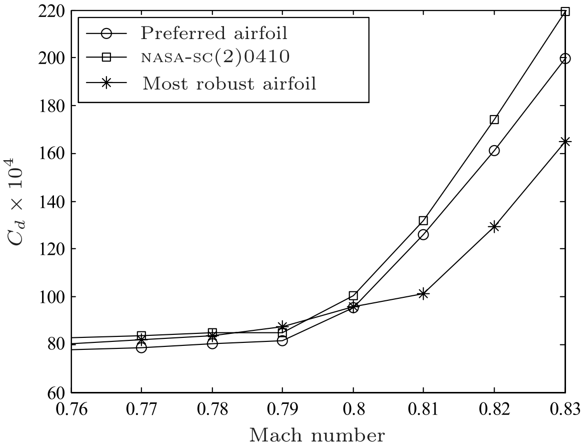
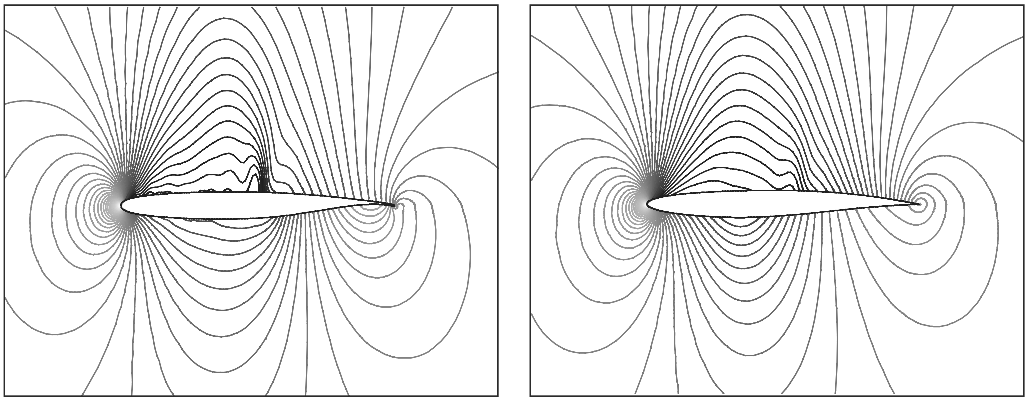


Fig. 20 Drag rise curves for abrupt changes in Mach number.



a) NASA-SC(2)0410 reference airfoil

b) Preferred airfoil

Fig. 19 Static pressure contours for $M = 0.79$ design condition.

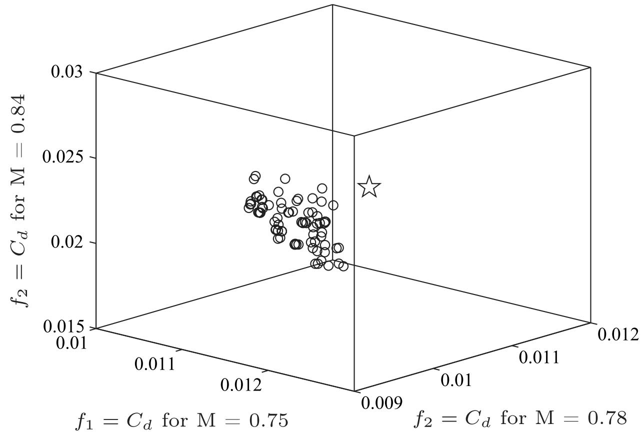


Fig. 21 Final Pareto front for multipoint transonic design.

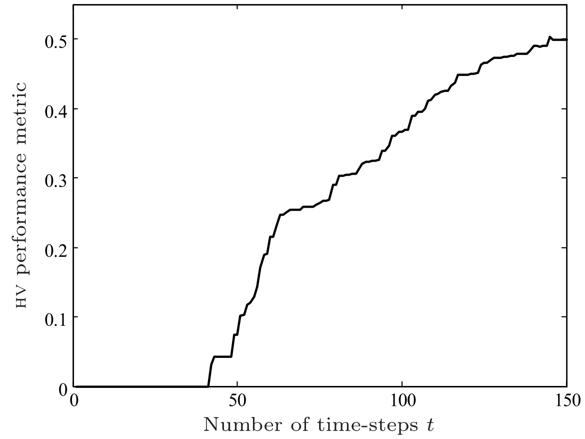


Fig. 22 HV performance curve for multipoint transonic design.

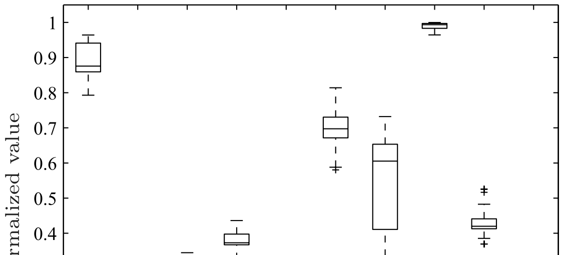


Fig. 23 Box plot of final Pareto-optimal design variables.

Table 4 Performance comparison of the preferred airfoil against NASA-SC(2) reference airfoils

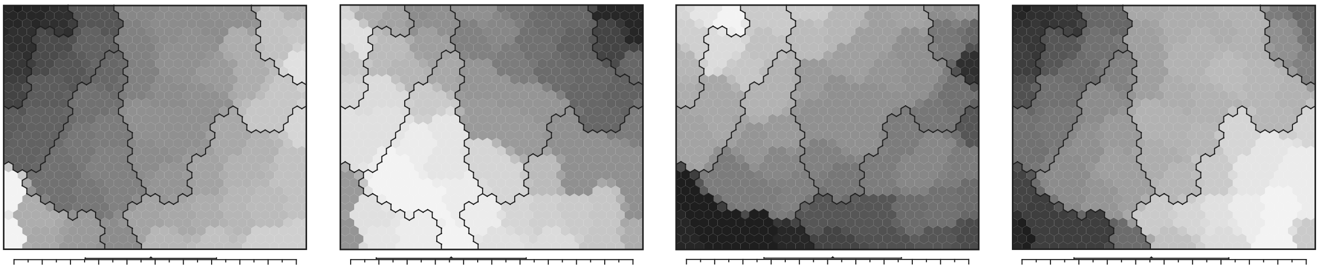
Condition	Reference	$C_d \times 10^4$	Angle α	Improvement
$M = 0.75$, $C_l = 0.7$	0712	111.9	0.357	5.1%
$M = 0.78$, $C_l = 0.6$	0610	96.3	−0.369	8.0%
$M = 0.84$, $C_l = 0.4$	0410	227.05	−0.926	3.4%

From the HV performance curve shown in Fig. 22, candidate solutions are not observed to enter within the preferred region until $t = 42$. To illustrate where the Pareto-optima are located within the landscape, a box plot is generated of the Pareto-optimal solutions in Fig. 23. The design space is normalized to $[0, 1]^n$ for best visualization. Figure 23 indicates that for all solutions, the leading-edge radius r_{LE} is situated at the upper bound of the permitted design space. The high r_{LE} shifts the pressure peak toward the leading edge and helps to maintain nonexcessive flow acceleration that results in decreased shock losses. This is also evident by the mild upper-surface curvature $z_{xx_{\mathrm{up}}}$ recorded, resulting in a relatively flat upper surface. The trailing-edge direction α_{TE} is observed to increase to generate sufficient positive camber in the aft section to meet the required lift constraints. This is coupled with a mild increase in the trailing-edge curvature. The most active variable is $z_{xx_{\mathrm{lo}}}$, which has the largest design range. Slight deviations in performance at various conditions are hence primarily attributed to the curvature of the lower surface.

The algorithm has been successful in obtaining solutions that exhibit improved aerodynamic performance over each reference airfoil in their respective operating conditions, while satisfying constraints. A similar SOM analysis is conducted to visualize tradeoffs between the design objectives and their respective resemblance to the reference-point compromise. A two-dimensional representation of the data is presented in Fig. 24, organized by seven SOM-ward clusters. For this case study, it is not immediately recognizable as to which objective holds the highest priority, suggesting that the reference point places a similar weighting on all design objectives. However, by placing more emphasis on a particular design objective, the deviation from the preferred compromise is immediately identified by closely observing the SOM charts.

As per the previous case study, our most preferred solution is selected as that which provides the most resemblance in compromise to $\bar{z}$. Table 4 compares the results obtained from the preferred airfoil against the NASA-SC(2) reference airfoils.

The preferred solution provides significant improvement across all three design conditions. If the designer were inclined toward a particular design condition, the solution spread provides the flexibility to select another suitable airfoil, one that perhaps yields greater improvement in a preferred design condition. Figures 25–27 compare the preferred airfoil against all three reference airfoils, with the respective C_p distributions. For the first two design conditions, the location of the shock is shifted forward in comparison with the NASA-SC(2) airfoils, however, the magnitude is significantly weaker, which is evident from the improvement in drag obtained. The third design condition is shown to produce a large pressure



a) SOM colored by f_1

b) SOM colored by f_2

c) SOM colored by f_3

d) SOM colored by d_z

Fig. 24 SOM charts to visualize optimal tradeoffs between objectives (lighter shading denotes value minima).

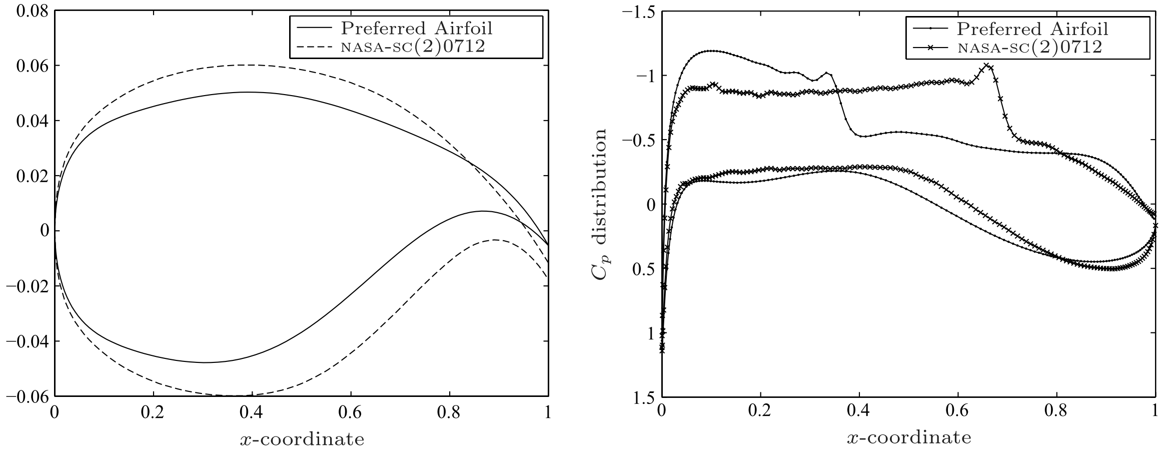


Fig. 25 Comparison of preferred airfoil with NASA-SC(2) 0712 for $M = 0.75$, $C_l = 0.7$.

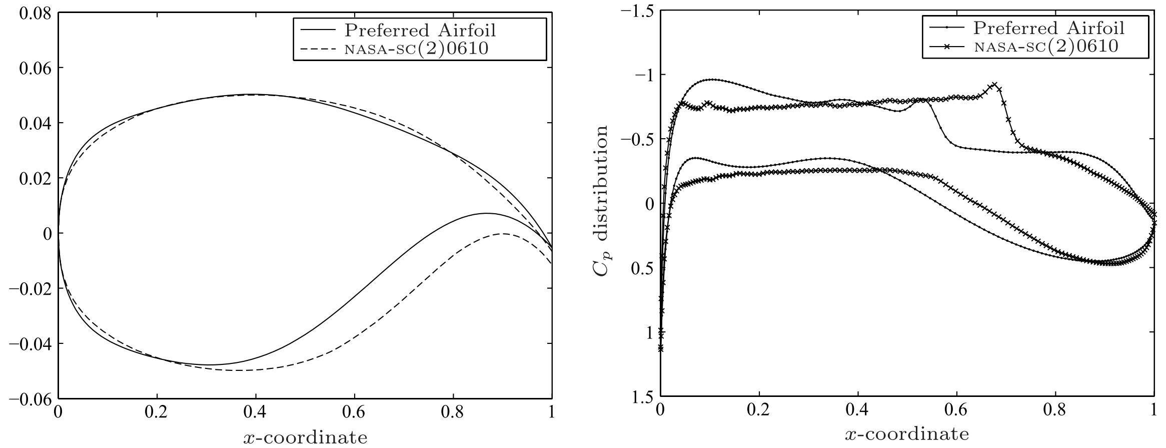


Fig. 26 Comparison of preferred airfoil with NASA-SC(2)0610 for $M = 0.78$, $C_l = 0.6$.

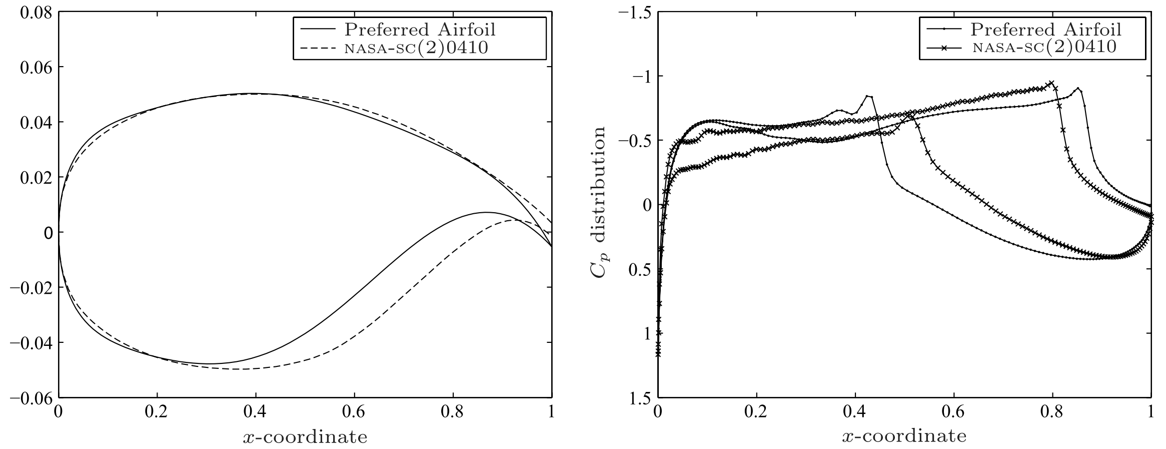


Fig. 27 Comparison of preferred airfoil with NASA-SC(2)0410 for $M = 0.84$, $C_l = 0.4$.

differential in the aft section, causing the moment constraint to be active. This poor design could be improved by placing more stringent priority on the third design condition.

V. Conclusions

This paper presented an efficient strategy to incorporate designer-driven preferences into a multi-objective particle swarm algorithm for airfoil design. Particles in the swarm communicate and coordinate themselves within the design space through distributed learning. A reference point is stipulated by the designer, which can ideally be based on an existing or target design. The reference-point module provides an effective guidance mechanism for the swarm, by selecting candidates for swarm leaders based on an inexpensive reference-point distance metric. This results in a final set of Pareto-optimal solutions that provide the most resemblance in compromise to the reference point. The increase in computational efficiency is attributed to the fact that the framework uses the knowledge possessed by the designer. The solutions obtained are hence reflective of the designer's preferred interest.

The optimization framework was applied to the design of transonic airfoils over a multi-operational spectrum. Experiments were conducted using a low-fidelity Euler solver. The two algorithms were compared using a hypervolume performance metric that provides a measure of both solution spread and convergence. The interactive UP-MOPSO algorithm proved to be more proficient in convergence over the unguided algorithm and provided greater precision in the exploitation of individual solutions. The UP-MOPSO algorithm was also applied to a higher-fidelity design case study, incorporating a Reynolds-averaged Navier–Stokes (RANS) flow solver. The reference points were ideally selected as an existing supercritical airfoil, in an attempt to improve on the performance characteristics of the airfoils, while maintaining a similar level of compromise between the design objectives. Simulations were performed in parallel using a high-performance cluster. Final results were observed to offer significant improvement over existing transonic designs for a range of operating conditions. The addition of user preferences was also observed to provide an intuitive means of selecting the most preferred solution. A visual tradeoff analysis using self-organizing maps was applied to screen through the identified solutions and quantify their resemblance to the reference-point compromise.

While the algorithm presented in this paper proved effective over the defined experimental test cases, no mathematical proofs of convergence are available at this time, nor has the performance of this algorithm over other interactive methods been examined, potentially warranting more research in this direction. Further studies could also evaluate alternative stopping criteria and the performance of the algorithm on larger-scale optimization problems. Further studies could also be conducted on performing preoptimization SOM analyses, to distinguish the correlation of the design space with the reference point, by use of the distance metric. This will allow for easy extraction of information from the design space, since the reference-point distance metric is a means of ranking or assessing the performance of a design using a single scalar, instead of an array of objective values. Such a visualization technique has the capacity to significantly aid the optimization process and will be a subject for future research.

Acknowledgments

The authors would like to gratefully acknowledge Tanja Zauner of Viscovery Software GmbH for kindly providing a license of Viscovery SOMine 5.2 for this research and the Victorian Partnership for Advanced Computing for assistance in the parallel implementation of the algorithm. Special thanks to Manas Khurana of the Sir Lawrence Wackett Aerospace Centre and to Xiaodong Li of the School of Computer Science and Information Technology. The authors would also like to gratefully acknowledge the feedback provided by András Sóbester from the University of Southampton.

References

[1] Keane, A. J., and Nair, P. B., *Computational Approaches for Aerospace Design: The Pursuit of Excellence*, Wiley, New York, 2005.

[2] Andersson, J., "A Survey of Multiobjective Optimization in Engineering Design," Linkping Univ., Dept. of Mechanical Engineering, TR LiTH-IKP-R-1097, Linkping, Sweden, 2000.

[3] Sobieszczanski-Sobieski, J., and Haftka, R. T., "Multidisciplinary Aerospace Design Optimization: Survey of Recent Developments," *Structural and Multidisciplinary Optimization*, Vol. 14, No. 1, 1997, pp. 1–23.
doi:10.1007/BF01197554

[4] Carrese, R., Winarto, H., and Watmuff, J., "User-Preference Particle Swarm Algorithm for Airfoil Design Architecture," *27th International Congress of the Aeronautical Sciences*, Nice, France, 2010.

[5] Wickramasinghe, U. K., Carrese, R., and Li, X., "Designing Airfoils Using a Reference Point Based Evolutionary Many-Objective Particle Swarm Optimization Algorithm," *IEEE World Congress on Computational Engineering*, Barcelona, Spain, 2010.

[6] Khurana, M. S., and Winarto, H., "Development and Validation of an Efficient Direct Numerical Optimisation Approach for Airfoil Shape Design," *The Aeronautical Journal*, Vol. 114, No. 1160, 2010, pp. 611–628.

[7] Ray, T., and Tsai, H. M., "Swarm Algorithm for Single and Multiobjective Airfoil Design Optimization," *AIAA Journal*, Vol. 42, No. 2, 2004, pp. 366–373.
doi:10.2514/1.9099

[8] Song, W., Keane, A., Eres, H., Pound, G., and Cox, S., "Two Dimensional Airfoil Optimisation Using CFD in a Grid Computing Environment," *Euro-Par 2003 Parallel Processing*, Springer, Berlin, 2004.

[9] Quagliarella, D., and Vicini, A., "Viscous Single and Multicomponent Airfoil Design with Genetic Algorithms," *Finite Elements in Analysis and Design*, Vol. 37, 2001, pp. 365–380.
doi:10.1016/S0168-874X(00)00053-6

[10] Deb, K., *Multi-Objective Optimization Using Evolutionary Algorithms*, Wiley, New York, 2001.

[11] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," Institute for Computer Applications in Science and Engineering, TR TR-97-9, Hampton, VA, 1997.

[12] Epstein, B., and Peigin, S., "Efficient Approach for Multipoint Aerodynamic Wing Design of Business Jet Aircraft," *AIAA Journal*, Vol. 45, No. 11, 2007, pp. 2612–2621.
doi:10.2514/1.29307

[13] Nemec, M., Zingg, D. W., and Pulliman, T. H., "Multipoint and Multi-Objective Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 42, No. 6, 2004, pp. 1057–1065.
doi:10.2514/1.10415

[14] Engelbrecht, A. P., *Fundamentals of Computational Swarm Intelligence*, Wiley, New York, 2005.

[15] Kennedy, J., and Eberhart, R. C., *Swarm intelligence*, Morgan Kaufmann, San Mateo, CA, 2001.

[16] Sierra, M. R., and Coello Coello, C. A., "Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art," *International Journal of Computational Intelligence Research*, Vol. 2, No. 3, 2006, pp. 287–308.

[17] Grosan, C., Abraham, A., and Ishibuchi, H. (eds.), *Hybrid Evolutionary Algorithms*, Springer–Verlag, Berlin, 2007.

[18] Song, W., "Multiobjective Memetic Algorithm and Its Application in Robust Airfoil Shape Optimization," *Studies in Computational Intelligence*, Springer, Berlin, 2009.

[19] Kampolis, I. C., and Giannakoglou, K. C., "A Multilevel Approach to Single- and Multiobjective Aerodynamic Optimization," *Computer Methods in Applied Mechanics and Engineering*, Vol. 197, 2008, pp. 2963–2975.
doi:10.1016/j.cma.2008.01.015

[20] Vicini, A., and Quagliarella, D., "Airfoil and Wing Design Through Hybrid Optimization Strategies," *AIAA Journal*, Vol. 37, No. 5, 1999, pp. 634–641.
doi:10.2514/2.764

[21] Carrese, R., Winarto, H., and Li, X., "Integrating User-Preference Swarm Algorithm and Surrogate Modeling For Airfoil Design," 49th AIAA Aerospace Sciences Meeting, Orlando, FL, AIAA Paper 2011-1246, 2011.

[22] Emmerich, M., and Naujoks, B., "Single- and Multi-Objective Evolutionary Optimization Assisted by Gaussian Random Field Metamodels," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 4, 2006, pp. 421–438.
doi:10.1109/TEVC.2005.859463

[23] Jeong, S., Murayama, M., and Yamamoto, K., "Efficient Optimization Design Method Using Kriging Model," 42nd AIAA Aerospace Sciences Meeting, Reno, NV, AIAA Paper 2004-118, 2004.

[24] Ehrgott, M., and Gandibleux, X., *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, International Series in Operations Research and Management Science, Vol. 52, Kluwer Academic, Norwell, MA, 2002.

[25] Rachmawati, L., and Srinivasan, D., "Preference Incorporation in Multi-Objective Evolutionary Algorithms: A Survey," *IEEE Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, 2006, pp. 962–968.

[26] Fonseca, C. M., and Fleming, P. J., "Multiobjective Optimization and Multiple Constraint Handling With Evolutionary Algorithms—Part I: A Unified Formulation," *IEEE Transactions on Systems, Man, and Cybernetics*, Pt. A, Vol. 28, No. 1, 1998, pp. 26–37.
doi:10.1109/3468.650319

[27] Thiele, L., Miettinen, P., Korhonen, P. J., and Molina, J., "A Preference-Based Interactive Evolutionary Algorithm for Multobjective Optimization," Helsinki School of Economics, Rept. W-412, Helsinki, Finland, 2007.

[28] Deb, K., and Kumar, A., "Interactive Evolutionary Multi-Objective Optimization and Decision-Making Using Reference Direction Method," *Genetic and Evolutionary Computation Conference*, ACM Press, New York, 2007, pp. 781–788.

[29] Deb, K., and Sundar, J., "Reference Point Based Multi-Objective Optimization Using Evolutionary Algortihms," *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2006, pp. 635–642.

[30] Wickramasinghe, U. K., and Li, X., "Integrating User Preferences with Particle Swarms for Multi-Objective Optimization," *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2008, pp. 745–752.

[31] Bratton, D., and Kennedy, J., "Defining a Standard for Particle Swarm Optimization," *Proceedings of the IEEE Swarm Intelligence Symposium*, IEEE Press, Piscataway, NJ, 2007, pp. 120–127.

[32] Clerc, M., and Kennedy, J., "The Particle Swarm—Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 1, 2002, pp. 58–73.
doi:10.1109/4235.985692

[33] Sierra, M. R., and Coello Coello, C. A., "Improving PSO-Based Multi-Objective Optimization Using Crowding, Mutation and ϵ-Dominance," *Lecture Notes in Computer Science*, Vol. 3410, 2005, pp. 505–519.
doi:10.1007/978-3-540-31880-4_35

[34] Wickramasinghe, U. K., and Li, X., "Using a Distance Metric to Guide PSO Algorithms for Many-Objective Optimization," *Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2009, pp. 667–674.

[35] Raquel, C. R., and Naval, P. C., "An Effective Use of Crowding Distance in Multiobjective Particle Swarm Optimization," *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, ACM Press, New York, 2005, pp. 257–264.

[36] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002, pp. 182–197.
doi:10.1109/4235.996017

[37] Zitzler, E., Deb, K., and Thiele, L., "Comparison of Multiobjective Evolutionary Algorithms: Empirical Results," *Evolutionary Computation*, Vol. 8, 2000, pp. 173–195.
doi:10.1162/106365600568202

[38] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E., "Scalable Test Problems for Evolutionary Multi-Objective Optimization," *Evolutionary Multiobjective Optimization (EMO): Theoretical Advances and Applications*, Springer, New York, 2005, pp. 105–145.

[39] FLUENT, Software Package, Ver. 6.3, ANSYS, Inc., Canonsburg, PA, 2006.

[40] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, 1992.

[41] MATLAB, Software Package, Ver. 7.9.0 R2009b, The Mathworks, Inc., Natick, MA, 2009.

[42] Song, W., and Keane, A. J., "A Study of Shape Parameterisation Methods for Airfoil Optimisation," 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, NY, AIAA Paper 2004-4482, 2004.

[43] Painchaud-Ouuellet, S., Tribes, C., Trepanier, J., and Pelletier, D., "Airfoil Shape Optimization Using a Nonuniform Rational B-Splines Parameterization Under Thickness Constraint," *AIAA Journal*, Vol. 44, No. 10, 2006, pp. 2170–2178.
doi:10.2514/1.15117

[44] Hacioglu, A., and Ozkol, I., "Transonic Airfoil Design and Optimisation by Using Vibrational Genetic Algorithm," *Aircraft Engineering and Aerospace Technology*, Vol. 75, No. 4, 2003, pp. 350–357.

[45] Lépine, J., Guibault, F., Trépanier, J. Y., and Pépin, F., "Optimized Nonuniform Rational B-Spline Geometrical Representation for Aerodynamic Design of Wings," *AIAA Journal*, Vol. 39, No. 11, 2001, pp. 2033–2041.
doi:10.2514/2.1206

[46] Sobieczky, H., "Parametric Airfoils and Wings," *Notes on Numerical Fluid Mechanics*, Vol. 68, Vieweg Verlag, 1998, pp. 71–88.

[47] Jahangirian, A., and Shahrokhi, A., "Inverse Design of Transonic Airfoils Using Genetic Algorithms and a New Parametric Shape Model," *Inverse Problems in Science and Engineering*, Vol. 17, No. 5, 2009, pp. 681–699.
doi:10.1080/17415970802417037

[48] Emmerich, M., Beume, N., and Naujoks, B., "An EMO Algorithm Using the Hypervolume Measure as Selection Criterion," *Lecture Notes in Computer Science*, Vol. 3410, Springer, New York, 2005, pp. 62–76.

[49] Kohonen, T., *Self-Organizing Maps*, Springer, Berlin, 1995.

[50] Jeong, S., Chiba, K., and Obayashi, S., "Data Mining for Aerodynamic Design Space," *Journal of Aerospace Computing, Information, and Communication*, Vol. 2, 2005, pp. 452–469.
doi:10.2514/1.17308