USER-PREFERENCE PARTICLE SWARM ALGORITHM FOR AIRFOIL DESIGN ARCHITECTURE

R. Carrese, H. Winarto, J. Watmuff
Royal Melbourne Institute of Technology, Bundoora, VIC, 3083, Australia

Keywords: *airfoils, CFD, optimization, swarm intelligence, user-preferences*

Abstract

Airfoil optimization frameworks rely on extensive computational resources to provide the most feasible design for the required flight condition. Extending the framework to encompass multiple design criteria adds further complexity due to the conflicting nature of the design objectives. In this paper, we propose the integration of designer-driven preferences into a multi-objective particle swarm heuristic to efficiently navigate the design space. This results in a highly accurate and proficient rate of convergence, since only preferred regions of the Pareto landscape are explored. The algorithm is introduced and applied to a typical transonic airfoil design scenario where the objective value is provided by a Reynolds-averaged Navier-Stokes solver. Experimental case-studies are presented, which highlight the computational efficiency of the user-preference algorithm.

1 Introduction

Progress in automated optimization and numerical modelling has led to the development of innovative engineering design frameworks. In recent times, focus has shifted towards the integration of multiple disciplines and objectives. High-fidelity engineering design problems are not easily managed, since the objective solver is computationally intensive, and the design landscape is generally multi-modal [1]. It may also prove difficult to screen through all possible design candidates, due to the conflicting nature of the objectives. Identifying and screening all possible candidates can be avoided if the designer has a reasonable understanding of the preferred level of compromise between the design objectives. This could be based on the designer's own refined knowledge of the problem domain, or data obtained from existing solutions [2]. This research draws on the recent progress achieved in interactive optimization to develop a designer-driven airfoil aerodynamic design synthesis.

For airfoil shape optimization, intelligent search algorithms are used to screen through candidate designs, which are ranked by Computational Fluid Dynamics (CFD) solvers. Classical optimization techniques, such as the gradient-based methods, have been successfully applied to airfoil design [3, 4]. However, such methods are restricted since they generally require that the objective function satisfy continuity and derivability conditions. Furthermore, since only one candidate solution is progressed, several runs are required to generate a set of Pareto-optimal solutions. By contrast, population-based Evolutionary Multi-Objective (EMO) algorithms are able to search for all Pareto-optimal solutions in one single run, providing the designer full flexibility in selecting the most appropriate solution [5, 6, 7]. In this paper, an optimization algorithm based on the Particle Swarm Optimization (PSO) heuristic is proposed. PSO is a fairly recent addition to the existing list of evolutionary methods, and is derived from the choreography of bird flock [8]. Particles identify and exploit promising areas of the design space by learning from previous experience and emulating the success of other particles. PSO has proven to be an effective technique to facilitate single-objective optimization problems, and has since gained rapid popularity in multi-objective optimization [9].

Despite the advantages offered by EMO, they may become computationally intensive since they are not guided by the differential landscape of the objective function. Furthermore, the process may converge pre-maturely if the objective landscape is multi-modal or the design space is large. Conventional EMO algorithms perform an 'unguided' search of the design space, subsequently providing a feasible set of solutions to the designer. Interactive methods incorporating the designer require various levels of user-input to obtain preferred solutions [10]. These classical methods are often disregarded since the required user-inputs (e.g. targets or bias weights) are generally not known in advance [5, 11]. There has however been increasing interest in coupling classical interactive methods to EMO algorithms as an intuitive way of specifying user-preferences [12, 13, 14]. Such methodologies focus all computing effort on preferred regions of the Pareto landscape, to find only solutions of interest to the designer. Interactive EMO have proven to be highly proficient in comparison to conventional EMO, especially in dealing with many objectives or multi-modality [2, 15]. This hybrid methodology is viewed as an improvement over traditional hybrid or memetic algorithms [16, 17], since additional function evaluations are not required to promote an accurate and accelerated rate of convergence.

In this paper, we present the framework of the user-preference module and discuss its integration into a multi-objective PSO algorithm (UP-MOPSO). The parameterization scheme and flow solvers adopted are described. A modified Hyper-Volume (HV) metric is utilized to highlight the convergence proficiency of the UP-MOPSO algorithm over a conventional unguided MOPSO algorithm. Conclusions are presented and avenues for future research are proposed. Future endeavours will concentrate on the adaptation of the framework to encompass multiple disciplines and surrogate assistance.

2 Direct Optimization Framework

An efficient strategy for the Direct Numerical Optimization (DNO) of transonic airfoils in cruise is presented (see Fig. 1). In transonic optimization, emphasis is placed on limiting shock drag losses and reducing shock-induced boundary layer separation at the desired operating condition. This generally occurs at the expense of excessive pitching moments due to aft loading and poor performance under off-design conditions. To improve on each of these design criteria, requires the formulation of a Multi-Objective Problem (MOP). In mathematical terms, a MOP is defined as:

$$\min_{\vec{x}\in X} f(\vec{x}) = \{f_1(\vec{x}),\ldots,f_m(\vec{x})\} \qquad (1)$$

where m is the number of objectives, and the solution $\vec{x} = \{x_1,\ldots,x_n\}$ is in the n-dimensional hypercube $X \in \mathbb{R}^n$, where X is the feasible search space. Solutions to Eq. 1 are classified Pareto-optimal, each providing a specific level of compromise between the objectives. To mathematically establish this concept, let two sets of

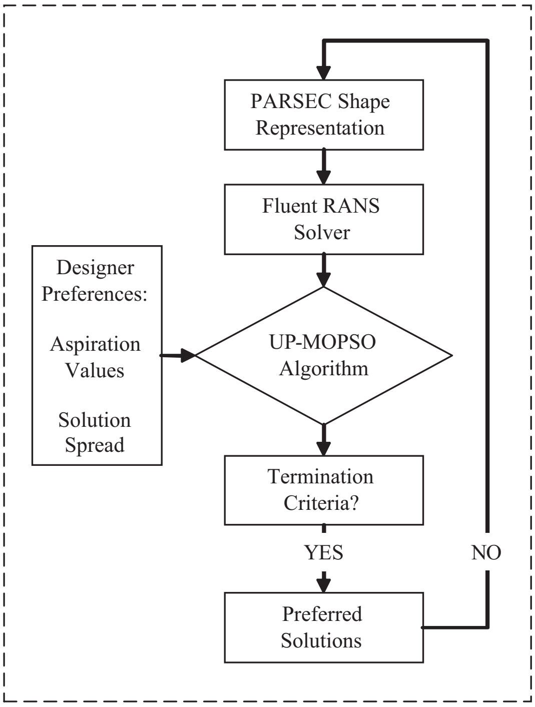

Fig. 1 Preference-based optimization framework for direct airfoil shape optimization

decision vectors, $\vec{a},\vec{b} \in X$. The decision vector $\vec{a}$ *strictly dominates* $\vec{b}$ (denoted by $\vec{a} \prec \vec{b}$) if,

$$\begin{aligned} \forall i \in (1,\dots,m) \;\; f_i(\vec{a}) \leq f_i(\vec{b}) \\ \text{and} \;\; \exists i : \;\; f_i(\vec{a}) < f_i(\vec{b}) \end{aligned} \tag{2}$$

It follows that a feasible solution $\vec{a}*$ is Pareto-optimal if there is no other feasible solution $\vec{a} \in X$ such that $f(\vec{a}) \prec f(\vec{a}*)$. The set of Pareto-optimal solutions is called the Pareto front.

The simplest way of determining the most 'preferred' solution is to aggregate the objectives into a single scalar through weights. However, this requires knowledge of the weight terms a priori which is often not possible. Furthermore, aggregation will not necessarily guarantee complete equivalence to the actual Pareto-optimal solution [11]. Alternatively, using the dominance criteria, a set of Pareto-optimal solutions can be determined in one single run.

2.1 UP-MOPSO Algorithm

The User-Preference Multi-Objective Particle Swarm Optimization (UP-MOPSO) algorithm applies the dominance criteria to explore the permissible 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 towards solutions of interest. This is an intuitive method to express the designer's preferred level of compromise, which could be based on some existing airfoil design. This results in a spread of 'preferred' Pareto-optimal designs which offer the most resemblance in compromise to the reference point. The additional guidance provided by the reference point constitutes a highly proficient convergence rate, coupled with greater precision in the exploitation of solutions, due to the locally focused search effort.

2.1.1 Particle Flight

The (MO)PSO heuristic utilizes a swarm of particles which are 'flown' through $X \in \mathbb{R}^n$. The i^{th} particle of the swarm is represented by the n-dimensional vectors $\vec{x}_i$ and $\vec{v}_i$, which are the particle position and velocity respectively. In this paper, the global best PSO by Clerc and Kennedy [18] has been adopted. The i^{th} particle is accelerated towards its personal best position, $\vec{p}_i$ and the global best position, $\vec{p}_g$ of the swarm. The velocity vector at time $t+1$ is a resultant of these two acceleration components.

$$\begin{aligned} \vec{v}_{i,(t+1)} = \chi\, [\vec{v}_i + c_1 r_1 (\vec{p}_i - \vec{x}_i) + \cdots \\ \cdots c_2 r_2 (\vec{p}_g - \vec{x}_i)]_t \end{aligned} \tag{3}$$

$$\vec{x}_{i,(t+1)} = \vec{x}_{i,(t)} + \vec{v}_{i,(t+1)} \tag{4}$$

Where the acceleration coefficients $c_1 = c_2 = \varphi/2$ and $\varphi = 4.1$. r_1 and r_2 are uniform random numbers $U(0,1)$ and provide the stochastic element of the algorithm. The constriction factor χ applies a dampening effect as to how far the particle explores within the search space, given as $\chi = 2/|2-\varphi-\sqrt{\varphi^2-4\varphi}|$.

2.1.2 Global Repository

Pareto-optimal solutions are all classified as equally optimal, hence there are several global best solutions or swarm *leaders*. At each time-step, the best representative front found by the particles is stored within an elitist archive. Solutions in the archive are non-dominated and have no record of constraint violation. To avoid over-population of the archive, a limited number of solutions N_b are permitted for entry. Each solution within the archive is identified as a possible global leader.

2.1.3 Mutation

Despite its favourable convergence rate, the performance of PSO may deteriorate when confronted with a highly multi-modal problem [11, 9]. This could lead to pre-mature convergence as a result of the swarm being trapped within a local front. In this algorithm, a non-uniform Gaussian mutation operator is applied to particles in the archive, with a 10% dimensional probability. To ensure that the mutation is non-destructive, mutated particles are cross-validated against existing non-dominated solutions. The mutation operator comes to effect when non-dominated solutions are not regularly updated. The percentage of mutated particles reduces as the size of the archive → N_b.

2.1.4 Integrating User-Preferences

To provide additional guidance in selecting candidates for global leadership from the elitist archive (see section 2.1.2), the reference point method is adopted, following the work of Wickramasinghe and Li [12]. This directs the swarm towards preferred regions of the Pareto front. A reference point $\bar{z}$ is used to construct a distance metric which is to be minimized over X,

$$\text{minimize } d_z = \max_{i=1:m} \{w_i \left(f_i(\vec{x}) - \bar{z_i} \right)\} \qquad (5)$$

where $\bar{z_i}$ is the i^{th} component of the reference point or the aspiration value to the ith objective. The term w_i is the weight value for the i^{th} fixed at one for the proceeding case studies. Reference points can ideally be placed in any feasible or infeasible region, since the designer generally has no prior knowledge of the Pareto front.

2.1.5 Algorithm Pseudo-Code

The UP-MOPSO algorithm combines the search strategy of (MO)PSO with the interactive reference point method. UP-MOPSO applies the dominance criteria concurrently with Eq. (5) to find a feasible set of Pareto-optimal solutions near the reference point. The spread of solutions along the Pareto front is controlled by δ. 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 the variance and evidently, as the value of δ escalates, the influence of the reference point location nullifies.

Solutions that are highly ranked according to Eq. (5) are selected as candidates for global leaders and assigned randomly to each particle in the population. If the solution limit of the repository is breached, lowest ranked solutions become candidates for replacement. Once the spread of solutions satisfies $\sigma(d_z) \leq \delta$, a crowding metric is used to select candidates for global leadership to promote an evenly distributed set of solutions. Particles in sparsely populated areas of the preferred region then become candidates for global leadership. Similarly, particles in densely populated areas become candidates for replacement.

The UP-MOPSO pseudo-code is presented in Fig. 2 below:

```
Initialize $\vec{x}$
Initialize $\vec{v}$
for $i = 1$ : Swarm Population, $N_s$ do
    Calculate objective $f_i$
    Calculate constraint $c_i$
    Calculate distance $d_z$
end for
Initialize $\vec{x}_i = \vec{x}$
Insert non-dominated solutions into archive
Sort archive according to minimum $d_z$
Initialize $t = 0$

while $t < t_{max}$ do
    $t = t + 1$
    Assign global leaders from 10% of archive
    Update $\vec{v}$ using Eq. 3
    Update $\vec{x}$ using Eq. 4
    for $i = 1 : N_s$ do
        Calculate objective $f_i$
        Calculate constraint $c_i$
        Calculate distance $d_z$
    end for
    Update archive
    if $size(\text{archive}) < N_b$ then
        Replace with max $d_z$
    end if
    if Mutation then
        Re-Update archive
    end if
    if $\sigma(d_z) > \delta$ then
        Sort archive according to minimum $d_z$
    else if $\sigma(d_z) \leq \delta$ then
        Sort archive according to crowding
    end if
end while
```

Fig. 2 UP-MOPSO algorithm pseudo-code

Figs. 3 and 4 demonstrate the UP-MOPSO algorithm applied to benchmark mathematical test functions. The ZTD3 [19] test function has a discontinuous Pareto landscape, which is representative of aerodynamic design problems. The

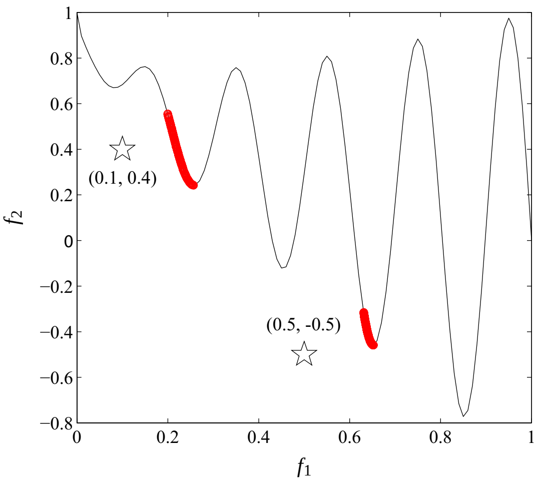

Fig. 3 UP-MOPSO on ZTD3 test problem

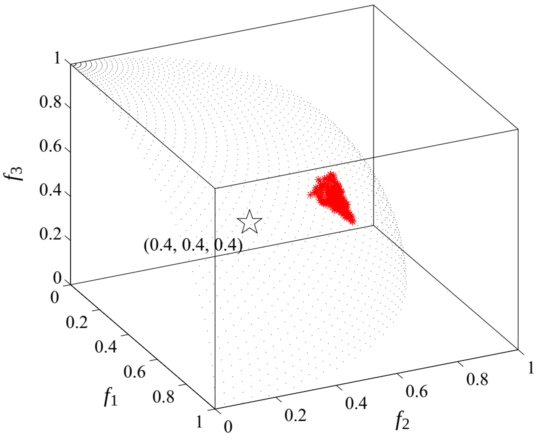

Fig. 4 UP-MOPSO on DTLZ2 test problem

DTLZ2 [20] test function is multi-modal 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 which are closest to the preferred regions reside on the Pareto fronts.

2.2 Computational Flow Solver

The finite volume code Fluent [21] is utilized to assign objective values. A pressure-based numerical procedure is adopted with third-order spatial discretization. The momentum equations and pressure-based continuity equation are solved concurrently, with the Courant-Friedrichs-Lewy (CFL) number set at 200. The Spalart-Allmaras closure equation [22] is applied assuming turbulent flow over the entire airfoil.

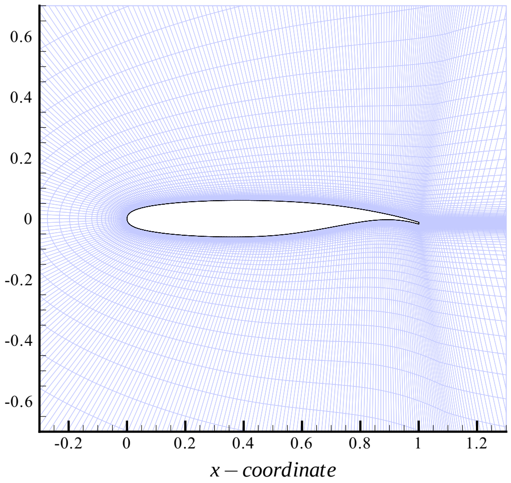

Fig. 5 RANS grid for transonic optimization

A C-type grid, stretching 25 chord lengths aft and normal of the airfoil section is used for the Fluent simulation. 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 abverage y-plus value of 120. A Full MultiGrid (FMG) Initialization scheme is employed, with coarsening of the grid to 30 cells. In the FMG initialization process, the Euler equations are solved using a first-order spatial discretization to obtain an approximation of the flowfield before submitting to the full iterative calculation.

2.3 Shape Representation

Shape representation is perhaps the most important contributing factor to an optimization framework, since it effectively defines the multimodality and topology of the design space [23]. In this research, we have opted for the use of the PARSEC method [24]. This method has the advantage of providing strict control

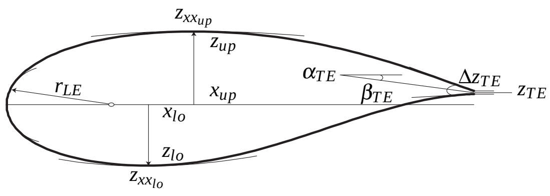

Fig. 6 PARSEC airfoil representation

over the airfoil geometry for imposing shape constraints. Illustrated in Fig. 6 are the basic eleven parameters that are used to completely define the profile geometry. They include the leading edge radius r_{LE}, the upper and lower crest locations $(x_{up}, z_{up}, x_{lo}, z_{lo})$ and curvatures $(z_{xx_{up}}, z_{xx_{lo}})$, trailing edge coordinate z_{TE} and thickness Δz_{TE}, trailing edge direction α_{TE} and wedge angle β_{TE}. The trailing edge coordinate z_{TE} and thickness variable Δz_{TE} are equal to zero and have been omitted.

The PARSEC method has been modified to include an additional trailing edge variable, $\delta_{\alpha_{TE}}$. Additional control over the trailing edge curvature is beneficial in order to reduce the probability of downstream boundary layer separation, giving rise to increased drag values. Further details of this modification are available in the original paper by Jahangirian and Shahrokhi [25].

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. This is achieved through inverse mapping of benchmark profiles that have been developed (either by experimental or computational methods) to perform favourably 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 [26]. A thickness constraint is explicitly defined through the parameter ranges as approximately 9.75% of chord.

Index	Variable	Upper	Lower
1	r_{LE}	0.0063	0.0151
2	α_{TE}	0.2405(-)	0.0026(-)
3	β_{TE}	0.0655	0.2618
4	x_{up}	0.3170	0.5250
5	z_{up}	0.0497	0.0683
6	$z_{xx_{up}}$	0.5135(-)	0.2393(-)
7	x_{lo}	0.2835	0.3418
8	z_{lo}	0.0603(-)	0.0478(-)
9	$z_{xx_{lo}}$	0.2535	0.8405
10	$\delta_{\alpha_{TE}}$	0.0080(-)	0.3696

Table 1 PARSEC ranges for transonic optimization

The ability to conform to transonic airfoils is best visualized by comparing resultant pressure distributions. The PARSEC method with additional trailing edge control, $\hat{z}$ was used to conform to the transonic RAE2822 airfoil, z. The PARSEC conformed airfoil was then subsequently analysed using the Fluent solver described in section 2.2. The Fluent simulation was compared to the experimental pressure distribution for the flight condition of M = 0.73, $\alpha = 3.19°$ (corrected to 2.79° for wind-tunnel effects) and Re = 6.5×10^6. This corresponds to case 9 of Cook et al. [27].

From visual inspection of Fig. 7, the error $(z^2 - \hat{z}^2)$ is very small in magnitude, particularly in the vicinity of the trailing edge. However, slight deviations in the airfoil geometry are am-

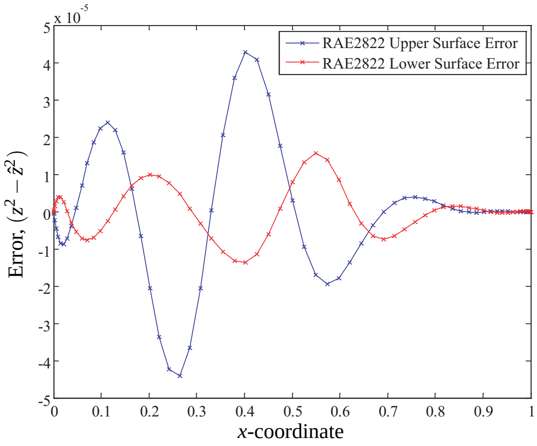

Fig. 7 PARSEC airfoil conformed to RAE2822

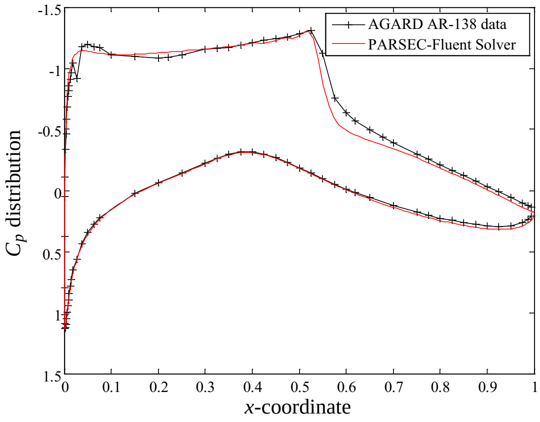

Fig. 8 Fluent solver compared to AR-138 data

plified when comparing the C_p distributions, as shown in Fig. 8. Despite the location of the shock shifting upstream, a reasonable agreement is obtained at the leading edge and trailing edge. Errors in the C_p distributions may also arise due to the fidelity of the solver, albeit both methods have been regarded as suitable for design optimization studies.

3 Experiments and Results

We present a series of airfoil design case studies. Firstly, experiments are conducted using a low-fidelity Euler solver with and without reference point specification. We hereby refer to the MOPSO algorithm with no reference point as the ‘unguided’ algorithm. The Euler solver is computationally inexpensive yet still able to capture important shock-flow phenomena. An example high-fidelity transonic case study adopting the Fluent RANS solver is then presented, for robust aerodynamic performance.

3.1 Performance Metric

There are several popular performance metrics that are used to monitor the solution spread and accuracy of EMO algorithms [19]. To monitor convergence and solution spread the Hyper-Volume (HV) metric described in Wickramasinghe, et al. [2] is used. The HV metric gives the total volume bounded by the points on the solution front and the nadir point. At the nadir point, all objectives are at their worst values simultaneously. When comparing the performance of two algorithms, the one which provides a better HV value is considered more proficient. Each experiment is repeated two times and the HV curve is a representation of the average progress of each algorithm. The HV metric provides a measurement on both the spread and closeness of the solution to the Pareto front.

3.2 Two-Objective Example

A swarm population of 100 is flown for 200 time-steps. The objectives are to minimize $f_1 = C_d$ for M = 0.75, $C_l \geq 0.70$ and $f_2 = C_d$ for M = 0.79, $C_l \geq 0.40$. An additional two decision variables are introduced for this problem, which are the angles $\alpha_{f_1}, \alpha_{f_2} \in [-1°, 1°]$. A constraint on the moment coefficient is imposed as $C_m^2 \leq 0.04$ for both operating conditions. The reference point is given as $\bar{z} = (0.0062, 0.0033)$ with a solution spread of $\delta = 0.001$.

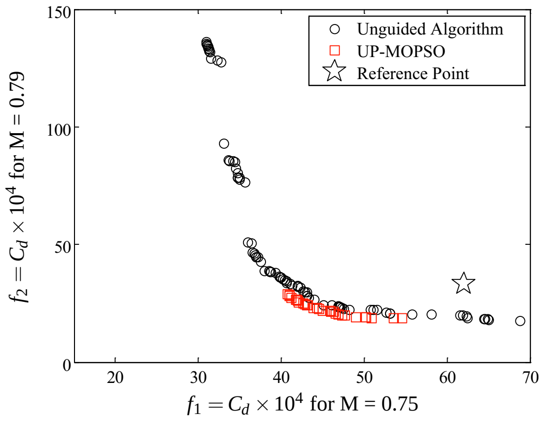

Fig. 9 Multi-point design Pareto front

It is observed from Fig. 9 that both algorithms have arrived to global Pareto front. However, the UP-MOPSO algorithm has a far superior convergence rate and provides a more precise set of solutions, as shown in Fig. 10.

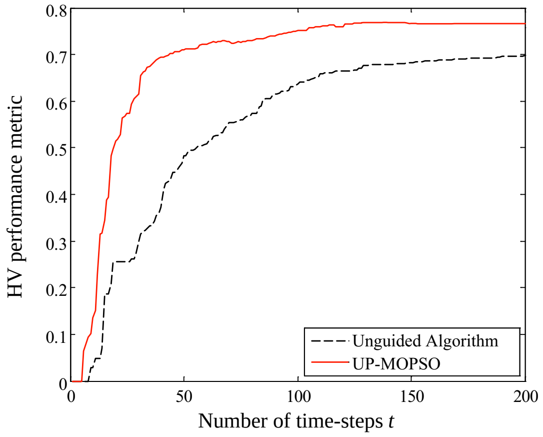

Fig. 10 Multi-point design HV curves

3.3 Three-Objective Example

A swarm population of 100 is flown for 200 time-steps. The objectives are given 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 which do not exhibit a high drag rise with an abrupt change in velocity, where $f_3 = \partial C_d/\partial(10 \cdot \mathrm{M})$ 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 $\delta = 1 \times 10^{-2}$.

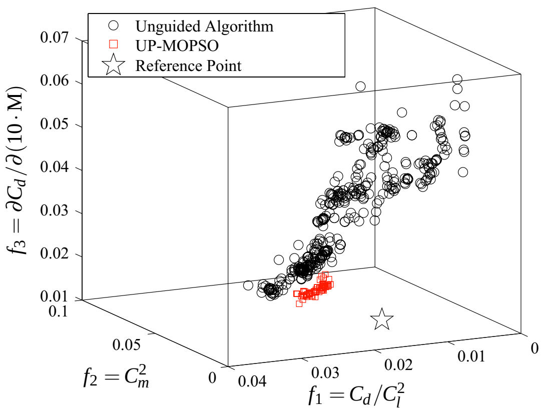

Fig. 11 Single-point design Pareto front

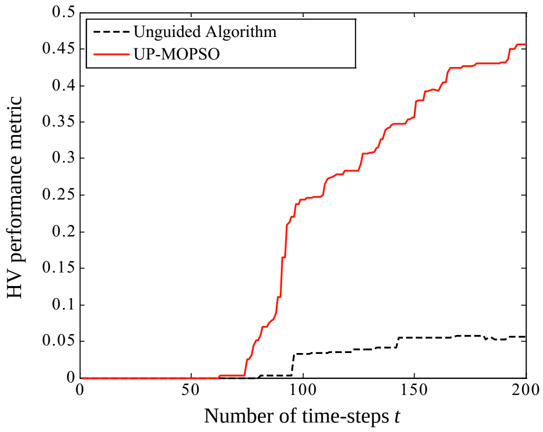

Fig. 12 Single-point design HV curves

It is clearly shown that the benefits of adopting the UP-MOPSO algorithm are apparent for three-objective problems. The unguided algorithm explores all regions of the Pareto landscape, and has converged pre-maturely to a sub-optimal front. On the contrary, the UP-MOPSO arrives at the global Pareto front (despite the infeasible location of $\bar{z}$) within the allocated number of time-steps. Since the HV metric (see Fig. 12) is a measure of both convergence and solution spread, the small increments in the HV value as $t \rightarrow 200$ are accredited to improvements in solution uniformity and spread.

3.4 High-Fidelity Design

We now apply the high-fidelity solver and the UP-MOPSO algorithm to a typical transonic design scenario. The algorithm is applied to the re-design of the NASA-SC(2) 0410 reference airfoil [28] 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 the drag-rise gradient $f_3 = \partial C_d/\partial(10 \cdot \mathrm{M})$ for an abrupt change in ΔM of 0.03. The abrupt change in velocity does not consider C_l correction. The angle α is varied within the range $[-1°, 1°]$. The aspiration values are obtained using the Fluent solver, given as $\bar{z} = (85.1, 0.0106, 0.0297)$.

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 2.3GHz quad core processors. The average computational time required per time-step is approximately 1.25 hours. The optimization sequence was terminated after $t = 125$ time-steps, since no collective improvement in the 75 obtained solutions was recorded. Fig. 13 shows the progress of the Pareto front over a number of time-steps. Solutions were not recorded to enter the preferred region until $t = 35$.

For the purposes of this paper, the most 'preferred' solution is ideally selected as the highest ranked solution according to Eq. (5). The concept of the solution spread offers flexibility to the designer if they were otherwise slightly inclined towards a specific objective. For this problem, the algorithm was successful in obtaining a solution which exhibits improvement over all three objectives compared to the reference solution (i.e. $\bar{z}$ is dominated). The preferred airfoil geometry is shown in figure 14 in comparison with the NASA(SC) 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 the improved C_m value.

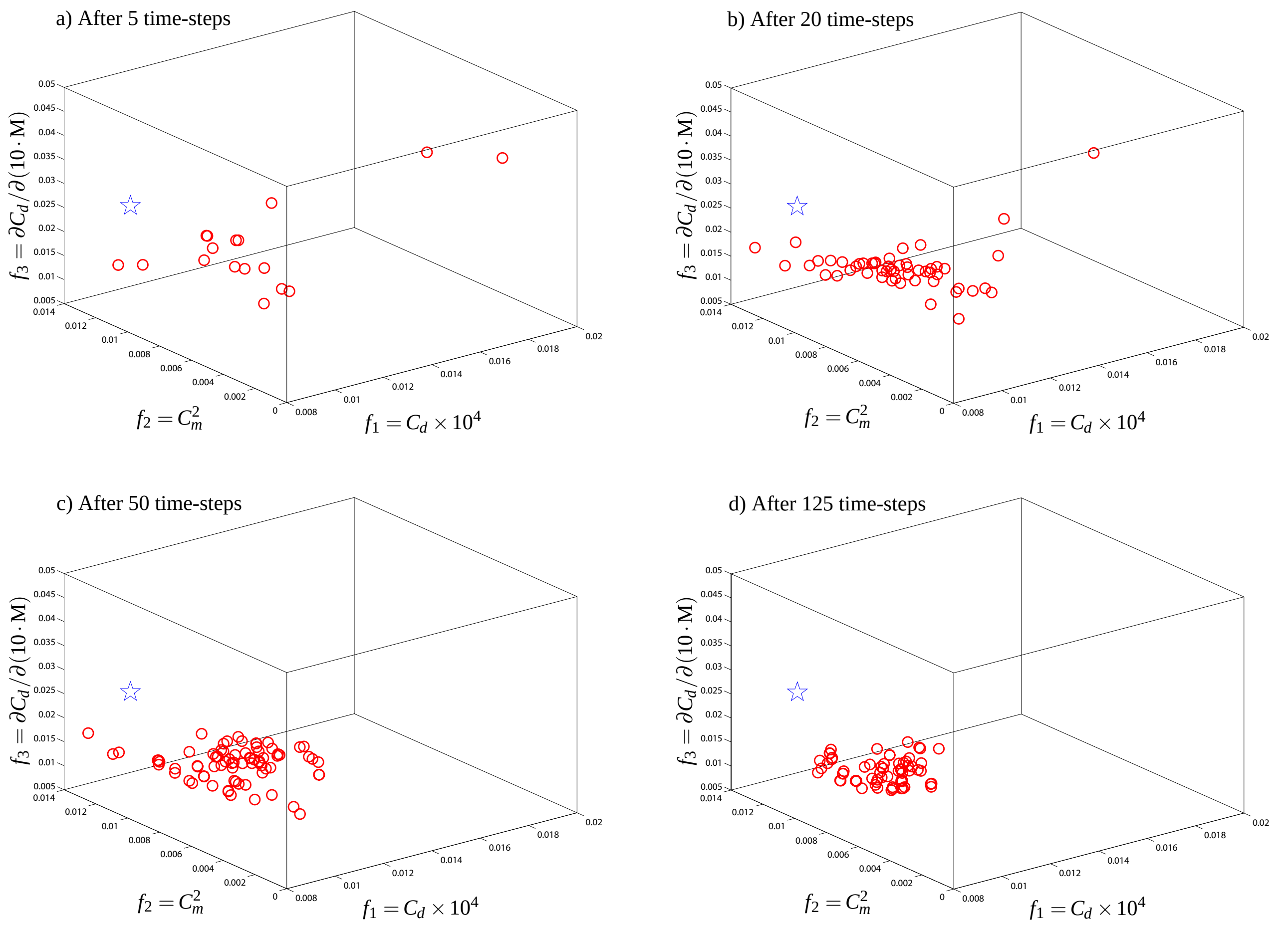

Fig. 13 Progress of Pareto front for robust single-point transonic design

The airfoil requires an angle $\alpha = 0.812°$ to satisfy the lift constraint. Values for the objective functions are given in table 2. The preferred airfoil has a significantly weaker shock in comparison to the 0410, highlighted by the C_p distributions of Fig. 15, and the static pressure contours of Fig. 16. This is predominantly due to the upper surface curvature, which does not produce excessive flow acceleration. An improvement of 4.1%, 30.7%, and 11.1% is recorded over the respective aspiration values. Fig.16(b) demonstrates the drag rise curves for the preferred solution compared with the 0410. Also shown is the solution which provides the most robust design (i.e. min f_3). The most robust design is clearly not obtained at the expense of poor performance at the design condition. The concept of the preferred region ensures that only solutions which slightly deviate from the compromise provided by $\bar{z}$ are identified.

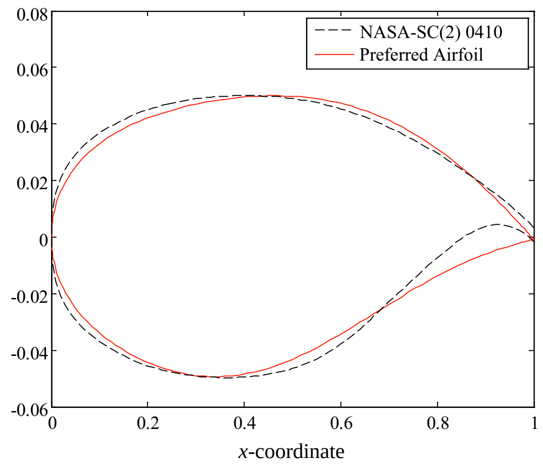

Fig. 14 Final preferred airfoil geometry

4 Conclusion

In this paper, we have presented a novel strategy to incorporate designer-driven preferences into a multi-objective particle swarm algorithm for

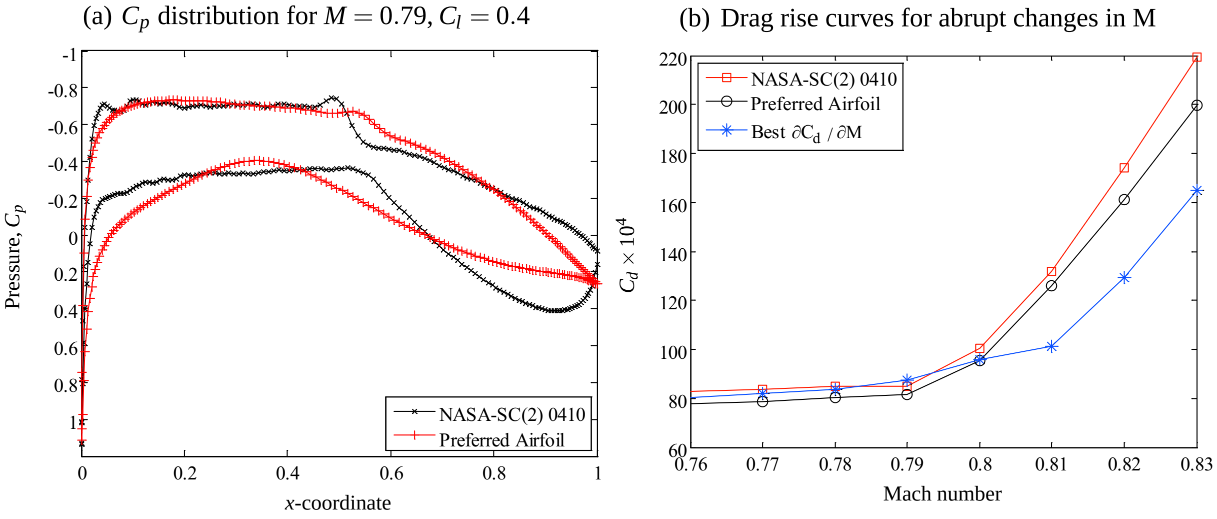

Fig. 15 Comparison between optimized airfoil and NASA-SC(2) 0410 airfoil

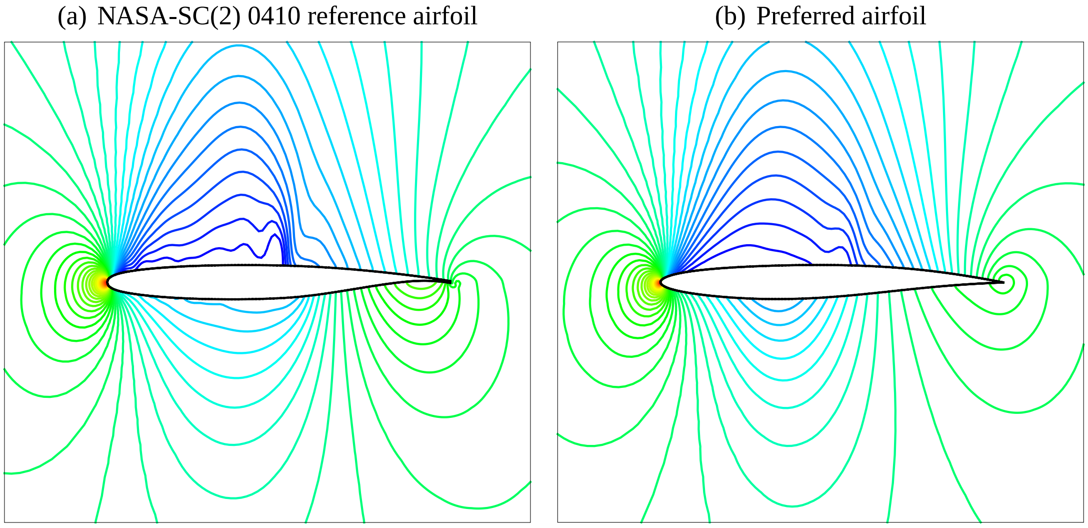

Fig. 16 Static pressure contours for M = 0.79 design condition

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

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

airfoil design. The user-preference module functions as a guidance mechanism for the swarm, by selecting candidates for global leaders based on an inexpensive reference point distance metric. This results in a final set of Pareto-optimal designs which provide the most resemblance in compromise to the reference point.

It was demonstrated through a series of direct optimization case-studies that the design framework is proficient in handling multi-modal landscapes with at least three conflicting objectives. The user-preference optimizer is observed to overcome problems that generally plague evolutionary multi-objective heuristics such as convergence over large design spaces and the exploitation of individual solutions. This is viewed as an improvement over traditional hybrid methodologies, since no additional function evaluations are required to promote an accurate and proficient rate of convergence. A high-fidelity experiment was performed in parallel incorporating a RANS solver. Results were observed to offer significant improvement over the reference airfoil. The adaptation of the reference point method to incorporate a solution spread offers the designer flexibility in selecting a solution which is more favourably inclined towards a specific design objective.

The next focus of this research is to adapt the framework for surrogate-assisted optimization. The user-preference module may establish an effective criterion for training response surface methods to model preferred regions of the objective landscape. This technique offers promise towards managing airfoil multi-disciplinary optimization problems which are restricted by a computational budget.

Acknowledgements

Manas Khurana of the Sir Lawrence Wackett Aerospace Centre, Upali Wickramasinghe and Xiaodong Li of the School of Computer Science and Information Technology and the Victorian Partnership for Advanced Computing (VPAC). 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*, John Wiley and Sons, 2005

[2] 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, 2010 (accepted)

[3] Anderson, W. K., and Venkatakrishnan, V., "Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation," Technical Report: TR-97-9, Institute for Computer Applications in Science and Engineering, 1997

[4] 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

[5] Deb. K., *Multi-objective optimization using evolutionary algorithms*, John Wiley and Sons, 2001

[6] 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

[7] 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

[8] Kennedy, J., and Eberhart, R. C., *Swarm intelligence*, Morgan Kaufmann Publishers Inc., 2001

[9] 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

[10] 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 Publishers, 2002

[11] Engelbrecht, A. P., *Fundamentals of computational swarm intelligence*, John Wiley and Sons, 2005

[12] 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, 2008, pp. 745–752

[13] 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, 2006, pp. 635–642

[14] Thiele, L., Miettinen, P., Korhonen, P. J., and Molina, J., "A preference-based interactive evolutionary algorithm for multobjective optimization," *Helsinki School of Economics* W-412, 2007

[15] 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, 2009, pp. 667–674

[16] Song, W., "Multiobjective memetic algorithm and its application in robust airfoil shape optimization," *Studies in Computational Intelligence*, Springer Berlin/Heidelberg, 2009

[17] Vicini, A., and Quagliarella, D., "Airfoil and wing design through hybrid optimization strategies," *AIAA Journal*, Vol. 37, No. 5, 1999, pp. 634–641

[18] 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

[19] Zitzler, E., Deb, K., and Thiele, L., "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, Vol. 8, 2000, pp. 173–195

[20] Deb, K., Thiele, L., et al., "Scalable test problems for evolutionary multi-objective optimization," *In Evolutionary Multiobjective Optimization (EMO): Theoretical Advances and Applications*, Springer, 2005, pp. 105-145

[21] ANSYS Fluent. Inc software, FLUENT 6.3 Documentation, 2006

[22] Spalart, P. R., and Allmaras, S. R., "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-0439, No. 1., 1992, pp. 5–21

[23] Song, W., and Keane, A. J., "A study of shape parameterisation methods for airfoil optimisation", *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Albany, New York, 2004

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

[25] 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

[26] Khurana, M. S., and Winarto, H., "Development and validation of an efficient direct numerical optimization approach for airfoil shape design," *The Aeronautical Journal*, 2010 (to be published)

[27] Cook, P. H., McDonald, M. A., and Firmin, M. C. P., Aerofoil RAE 2822 pressure distribution and boundary layer and wake measurements, *AGARD Advisory Report No. 138*, 1979.

[28] Harris, C. D., "NASA supercritical airfoils: A matrix of family-related airfoils," *Langley Research Center*, NASA-TP-2969, 1990

Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS2010 proceedings or as individual off-prints from the proceedings.