# Constrained Multi-Objective Aerodynamic Shape Optimization via Swarm Intelligence

Saúl Zapotecas Martínez
Shinshu University
Faculty of Engineering
4-17-1 Wakasato, Nagano,
380-8553, Japan
saul.zapotecas@gmail.com

Alfredo Arias-Montaño
IPN-ESIME
Departamento de Ingeniería
Aeronáutica
México, D.F. 07340, MEXICO
aarias@ipn.mx

Carlos A. Coello Coello
CINVESTAV-IPN (EVOCINV)
Departamento de
Computación
México, D.F. 07360, MEXICO
ccoello@cs.cinvestav.mx

## ABSTRACT

In this paper, we present a Multi-objective Particle Swarm Optimizer (MOPSO) based on a decomposition approach, which is proposed to solve Constrained Multi-Objective Aerodynamic Shape Optimization Problems (CMO-ASOPs). The constraint-handling technique adopted in this approach is based on the well-known $\varepsilon$-constraint method. Since the $\varepsilon$-constraint method was initially proposed to deal with constrained single-objective optimization problems, we adapted it so that it could be incorporated into a MOPSO. Our main focus is to solve CMO-ASOPs in an efficient and effective manner. The proposed constrained MOPSO guides the search by updating the position of each particle using a set of solutions considered as the global best according to both the decomposition approach and the $\varepsilon$-constraint method. Our preliminary results indicate that our proposed approach is able to outperform a state-of-the-art MOEA in several CMO-ASOPs.

## Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search.*

## Keywords

Constrained Multi-objective Optimization, Multi-objective Swarm Optimization, Aerodynamic Shape Optimization.

## 1. INTRODUCTION

Multi-objective Evolutionary Algorithms (MOEAs) have been successfully applied to the solution of a wide variety of problems in the fields of science and engineering [6, 7]. Although MOEAs were originally designed for solving unconstrained multi-objective optimization problems, most real-world applications require satisfying equality and/or inequality constraints. Generally, Constrained Multi-objective Optimization Problems (CMOPs) are difficult to solve, because finding feasible solutions may require substantial additional computational resources (i.e., an important number of extra objective function evaluations). In aeronautical systems design as well as in the design of propulsion system components, such as turbine engines, aerodynamics plays a key role. Thus, Aerodynamic Shape Optimization (ASO) is a crucial task, which has been extensively studied and developed. This discipline has recently benefited from the use of MOEAs, which have gained an increasing popularity in the last few years. For a detailed survey of applications of MOEAs in this field, the interested reader is referred to [3].

The simplicity and success of particle swarm optimization (PSO) algorithms, has motivated researchers to extend their use to solve Multi-objective Optimization Problems (MOPs) in the field of engineering and science, giving rise to the well-known Multi-Objective Particle Swarm Optimizers (MOPSOs). In recent years, the use of swarm intelligence applied to ASO problems has attracted the interest of several researchers. However, most of these approaches have been used to deal with single-objective ASO problems (see e.g. [20]) or with unconstrained MOPs (see e.g. [21]).

In this paper, we study the performance of a recent Multi-Objective Particle Swam Optimizer based on decomposition (called here DMOPSO[1]) [23], which was initially introduced for solving unconstrained MOPs. Here, we add to DMOPSO a constraint-handling mechanism based on the well-known $\varepsilon$-constraint method [19], which was proposed for single-objective optimization. The resulting approach is used to solve CMO-ASOs, which, are normally difficult to tackle in an efficient way, because of the high computational time that it is required to evaluate their objective functions. We argue that our proposed approach is a viable choice for solving CMO-ASOs in an effective and efficient way.

The remainder of this paper is organized as follows. In Section 2, we provide the basic concepts required for understanding the rest of the paper. Section 3 describes our proposed approach, including a detailed explanation of the adaptation of the $\varepsilon$-constraint method into DMOPSO in order to maintain a suitable balance between feasible solutions and convergence towards the Pareto optimal front. Section 4 presents the experimental study used for assessing the performance of our proposed algorithm. In Section 5, we provide a discussion of our results. Finally, in Section 6, we

*GECCO'14,* July 12–16, 2014, Vancouver, BC, Canada.
Copyright 2014 ACM 978-1-4503-2662-9/14/07 ...$15.00.
http://dx.doi.org/10.1145/2576768.2598372.

[1] The acronym DMOPSO (used in this paper) should not be confused with the Dynamic Multiobjective Particle Swarm Optimization (DMOPSO) presented in [11].

provide our conclusions and some possible paths for future research.

## 2. BASIC CONCEPTS

### 2.1 Constrained Multi-objective Optimization

A nonlinear Constrained Multi-objective Optimization Problem (CMOP) can be stated as (assuming minimization of all the objective functions):

$$\begin{array}{lll} \min & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T & \\ \text{s.t.} & g_i(\mathbf{x}) \leq 0, & i = 1, \dots, p \\ & h_j(\mathbf{x}) = 0, & j = 1, \dots, q \\ & L_i \leq x_i \leq U_i, & i = 1, \dots, n \end{array} \quad (1)$$

where $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbb{R}^n$ is an $n$ dimensional vector of decision variables, $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$ represent the $q$ inequality constraints and the $p$ equality constraints, respectively. $L_i$ and $U_i$ are the lower and upper bounds of each of the decision variables $x_i$. The vector $\mathbf{F}$ consists of $k$ objective functions $f_i$'s $(i = 1, \dots, k)$ to be minimized. The set of solutions that satisfy the constraints of problem (1) defines the feasible region $\Omega \subseteq \mathbb{R}^n$.

In order to describe the concept of optimality in which we are interested on, the following definitions are introduced [13]:

**Definition 1.** Let $\mathbf{x}, \mathbf{y} \in \Omega$, we say that $\mathbf{x}$ *dominates* $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) if and only if, $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $f_i(\mathbf{x}) < f_i(\mathbf{y})$ in at least one $f_i$ for all $i = 1, \dots, k$.

**Definition 2.** Let $\mathbf{x}^\star \in \Omega$, we say that $\mathbf{x}^\star$ is a *Pareto optimal* solution, if there is no other solution $\mathbf{y} \in \Omega$ such that $\mathbf{y} \prec \mathbf{x}^\star$.

**Definition 3.** The *Pareto optimal set PS* is defined by: $PS = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal solution}\}$, and the *Pareto optimal front PF* is defined as: $PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \mathcal{PS}\}$.

We thus wish to find the best possible *trade-offs* among the objectives, such that no objective can be improved without worsening another one. However, in this case, our solutions must also satisfy the constraints of problem (1). We are also interested in generating a set of solutions that are well-distributed along the Pareto front.

### 2.2 Decomposition of a Multi-objective Optimization Problem

It is well-known that a Pareto optimal solution to the problem (1), under certain conditions, could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the objective functions $f_i$'s. Many scalar approaches have been proposed to aggregate the objectives of an MOP. Among them, the Tchebycheff approach is one of most widely used methods reported in the specialized literature. In the following, we describe the Tchebycheff problem which is adopted in this study. Note however, that other scalarization approaches could also be easily coupled to this work–see for example those presented in [8, 13].

*Tchebycheff approach*

This approach transforms the vector of function values $\mathbf{F}$ into a scalar optimization problem which is of the form:

$$\begin{array}{ll} \min & g^{te}(\mathbf{x}|\mathbf{w}, \mathbf{z}) = \max_{1 \leq j \leq k}\{w_j|f_j(\mathbf{x}) - z_j|\} \\ s.t. & \mathbf{x} \in \Omega \end{array} \quad (2)$$

where $\Omega$ is the feasible region, $\mathbf{z} = (z_1, \dots, z_k)^T$, such that: $z_j = \min\{f_j(\mathbf{x})|\mathbf{x} \in \Omega\}$ for each $i = 1, \dots, k$ and $\mathbf{w} = (w_1, \dots, w_k)^T$ is a weight vector, i.e., $w_j \geq 0$ for all $j = 1, \dots, k$ and $\sum_{j=1}^k w_j = 1$.

For each Pareto optimal point $\mathbf{x}^\star$ there exists a weight vector $\mathbf{w}$ such that $\mathbf{x}^\star$ is the optimum solution of equation (2) and each optimal solution of equation (2) is a Pareto optimal solution of equation (1). An appropriate representation of the Pareto front could be reached by solving different scalarization problems. Such problems can be defined by a set of well-distributed weight vectors, which establish the search direction in the optimization process.

### 2.3 Constraint Violation Degree

The overall constraint violation $\phi(\mathbf{x})$ of a solution $\mathbf{x}$ can be given by the maximum of all constraints or the sum of all constraints.

$$\phi(\mathbf{x}) = \max\{\max_i\{0, g_i(\mathbf{x})\}, \max_j\{|h_j(\mathbf{x})|\}\} \quad (3)$$

$$\phi(\mathbf{x}) = \sum_i ||\max\{0, g_i(\mathbf{x})\}||^\alpha + \sum_j ||h_j(\mathbf{x})||^\alpha \quad (4)$$

where $\alpha$ is a positive number, i.e. $\alpha \in (0, +\infty)$. Equality constraints can be transformed into inequality constraints by using [5]:

$$|h_j(\mathbf{x}) - \varepsilon| \leq 0, j = 1, \dots, q \quad (5)$$

where $\varepsilon$ is a small real-value threshold.

For methods that do not require gradient information, it does not matter if equation (5) is non-differentiable. Therefore, a CMOP with equality constraints can be stated as a CMOP having only inequality constraints. Assuming that all constraints of problem (1) are inequality constraints, the constraint violation in equations (3) and (4) can be computed as:

$$\phi(\mathbf{x}) = \max_{1 \leq i \leq p}\{0, g_i(\mathbf{x})\} \quad (6)$$

$$\phi(\mathbf{x}) = \sum_{i=1}^{p} ||\max\{0, g_i(\mathbf{x})\}||^\alpha \quad (7)$$

Thus, we can say that without loss of generality, in this paper we consider only constraint functions of the form $g(\mathbf{x}) \leq 0$. Any equality constraint $h(\mathbf{x}) = 0$ needs to be transformed into an inequality constraint using equation (5).

### 2.4 The $\varepsilon$-Constraint Method

The $\varepsilon$-constraint method for single-objective optimization was proposed by Takahama and Sakai [19]. This constraint-handling technique adopts a lexicographic ordering with relaxation of the constraints. The basic idea is to define an $\varepsilon$ level comparison in order to state an order relation on pairs consisting of the objective function value and the constraint violation value, $(f(\mathbf{x}), \phi(\mathbf{x}))$. If the violation of a constraint is greater than 0, then the solution is not feasible and its value will be low. Since the feasibility of a solution $\mathbf{x}$ is more important than the minimization of the objective function, the $\varepsilon$ level comparisons are defined as a lexicographic order in which $\phi(\mathbf{x})$ precedes $f(\mathbf{x})$.

Let $f_1(f_2)$ and $\phi_1(\phi_2)$ be the function values and the constraint violation at a solution $\mathbf{x}^1(\mathbf{x}^2)$, respectively. Then, for any $\varepsilon$ satisfying $\varepsilon \geq 0$, the $\varepsilon$ level comparison $<_\varepsilon$ and $\leq_\varepsilon$ between $(f_1, \phi_1)$ and $(f_2, \phi_2)$ is defined as follows:

$$(f_1, \phi_1) <_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 < f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \tag{8}$$

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \tag{9}$$

In case of $\varepsilon = \infty$, the $\varepsilon$ level comparison $<_\infty$ and $\leq_\infty$ are equivalent to the ordinal comparison $<$ and $\leq$ between function values. Furthermore, the cases $\varepsilon = 0$, $<_0$ and $\leq_0$ are equivalent to the lexicographic order in which the constraint violation $\phi(\mathbf{x})$ precedes the function value $f(\mathbf{x})$.

## 2.5 The Framework of DMOPSO

The decomposition-based Multi-Objective Particle Swarm Optimizer (called here DMOPSO) [23], transforms a MOP into several scalarization problems. Therefore, an approximation of the Pareto front is obtained by solving the $N$ scalarization subproblems in which a MOP is decomposed.

Considering $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ as a well-distributed set of weight vectors, DMOPSO finds the best solution to each subproblem defined by each weight vector using the Tchebycheff approach. The objective function of the $j^{th}$ subproblem is then defined by $g^{te}(\mathbf{x}|\mathbf{w}^j, \mathbf{z})$, where $\mathbf{w}^j \in W$ and $\mathbf{z} = (z_1, \ldots, z_k)^T$ is the artificial utopian vector whose component $z_i$ is the minimum value found so far for the objective $f_i$.

At each generation, DMOPSO finds the best solution to each subproblem throughout the evolutionary process and maintains:

1. a population of $N$ particles $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, where $\mathbf{x}^i$ is the current solution to the $i^{th}$ subproblem; the velocity $\mathbf{v}^i$, the age $a^i$ and the personal best $\mathbf{x}^i_{pb}$ of each particle $\mathbf{x}^i$, for each $i = 1, \ldots, N$.

2. $F^1, \ldots, F^N$, where $F^i$ is the $F$-value of each particle $\mathbf{x}^i$, i.e., $F^i = \mathbf{F}(\mathbf{x}^i)$ for each $i = 1, \ldots, N$;

3. the global best set $G_{best} = \{\mathbf{g}^1, \ldots, \mathbf{g}^N\}$ which constitutes the set of particles considered to be the best for the $i^{th}$ subproblem along the search process.

**Algorithm 1:** General Framework of DMOPSO

```
Input:
a stopping criterion;
$N$: the number of the subproblems considered in $\varepsilon$DMOPSO;
$W$: a set of weight vectors $\{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$;
Output:
$P$: the final swarm of particles found by $\varepsilon$DMOPSO.
Initialize a swarm of $N$ particles $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$;
$\mathbf{z} = (+\infty, \ldots, +\infty)^T$;
foreach $\{1, \ldots, N\}$ do
    $F^i = \mathbf{F}(\mathbf{x}^i)$;
    $\mathbf{v}^i = \mathbf{0}$;
    $\mathbf{x}^i_{pb} = \mathbf{x}^i$;
    $a^i = 0$;
    If $f_j(\mathbf{x}^i) < z_j$ then $z_j = f_j(\mathbf{x}^i)$, for each $j = 1, \ldots, k$;
end
$G_{best} = P$;
while stopping criterion is not satisfied do
    foreach $i \in \{1, \ldots, N\}$ do
        Select randomly a solution $\mathbf{g}^i_{best}$ from $G_{best}$, such
        that: $\mathbf{x}^i \neq \mathbf{g}^i_{best}$;
        if $a^i < T_a$ then
            // Using $\mathbf{g}^i_{best}$ and $\mathbf{x}^i_{pb}$ do:
            Update velocity $\mathbf{v}^i$ (equation (10));
            Update position $\mathbf{x}^i$ (equation (11));
            $a^i = a^i + 1$;
        else
            Reset the particle $\mathbf{x}^i$ (equation (12));
            $\mathbf{v}^i = \mathbf{0}$;
            $a^i = 0$;
        end
        Repair bounds of the particle $\mathbf{x}^i$ (equation (13));
        Calculate $\mathbf{F}(\mathbf{x}^i)$;
        If $f_j(\mathbf{x}^i) < z_j$ then $z_j = f_j(\mathbf{x}^i)$, for each $j = 1, \ldots, k$;
        // Updating $\mathbf{x}^i_{pb}$:
        if $g^{te}(\mathbf{x}^i|\mathbf{w}^i, \mathbf{z}) < g^{te}(\mathbf{x}^i_{pb}|\mathbf{w}^i, \mathbf{z})$ or $a^i = 0$ then
            $\mathbf{x}^i_{pb} = \mathbf{x}^i$;
            $a^i = 0$;
        end
        // Using Algorithm 2:
        $G_{best} = update\_G_{best}(P \cup G_{best}, W)$;
    end
end
```

The general framework of DMOPSO is presented in Algorithm 1, where each particle in the swarm is evolved by updating both its velocity $\mathbf{v}^i$ and its position $\mathbf{x}^i$ according to the following equations:

$$\mathbf{v}^i = w\mathbf{v}^i + c_1 r_1(\mathbf{x}^i_{pb} - \mathbf{x}^i) + c_2 r_2(\mathbf{g}^i_{best} - \mathbf{x}^i) \tag{10}$$

and the new particle's position is updated according to the equation:

$$\mathbf{x}^i = \mathbf{x}^i + \mathbf{v}^i \tag{11}$$

where $w \geq 0$ represents the inertia factor, $c1, c2 \geq 0$ are the constraints on the velocity, $r_1, r_2$ are two random variables having a uniform distribution in the range $(0, 1]$, $\mathbf{v}^i, \mathbf{x}^i_{pb}$ and $\mathbf{g}^i_{best}$ represent the velocity, the personal best and the selected global best position for the $i^{th}$ particle, respectively.

If a particle does not improve its personal position in a flight cycle, then the particle increases (by one) its age. On the other hand, if a particle exceeds a certain (pre-defined) age threshold, the particle (including, its velocity, its age and its best personal) is reinitialized. DMOPSO employs a reinitialization mechanism based on a parametric probability density function, which involves the selected global best $\mathbf{g}^i_{best}$ and the personal best $\mathbf{x}^i_{pb}$ of the current particle $\mathbf{x}^i$. Therefore, the $j^{th}$ component of the new particle is reset according to the following equation:

$$x^i_j = N\left(\frac{g^i_{best,j} - x^i_{pb,j}}{2}, |g^i_{best,j} - x^i_{pb,j}|\right) \tag{12}$$

**Algorithm 2:** *update_$G_{best}(Q, W)$*

```
$T = Q$;
foreach $i = \{1, \ldots, N\}$ do
    $\mathbf{g}^i = \{\mathbf{x} | \min_{\mathbf{x} \in T} g^{te}(\mathbf{x}|\mathbf{w}^i, \mathbf{z})\}$, where $\mathbf{w}^i \in W$;
    $T = T \setminus \{\mathbf{x}\}$
end
$G = \{\mathbf{g}^1, \ldots, \mathbf{g}^N\}$;
return $G$;
```

where $N(m, \sigma)$ represents a random number using a Gaussian distribution with mean $m$ and sigma $\sigma$.

When a particle goes beyond the boundaries of the decision variables, this particle needs to be repaired. DMOPSO adopts the *deterministic back* mechanism [4] to repair solutions that are generated outside the allowable bounds. Therefore, the $j^{th}$ bound of the particle's position $\mathbf{x}^i = (x_1^i, \ldots x_n^i)^T$ and of its velocity $\mathbf{v}^i = (v_1^i, \ldots, v_n^i)^T$, are reestablished as follows:

$$\begin{array}{rcl} x_j^i & = & \begin{cases} x_j^L, & \text{if } x_j^i < x_j^L \\ x_j^U, & \text{if } x_j^i > x_j^U \end{cases}, \quad j = 1, \ldots, n \\ v_j^i & = & -\gamma v_j^i, \quad j = 1, \ldots, n. \end{array} \tag{13}$$

where $x_j^L$ and $x_j^U$ are the lower and upper bounds in the component $j^{th}$ (for each $j = 1, \ldots, n$) of the allowable decision variable values, respectively. In the same way than other MOPSOs [15, 14], DMOPSO adopts $\gamma = 1$.

In DMOPSO, a $(\mu + \lambda)$-selection mechanism is employed in order to obtain the global best set by using the current set of particles $P$ and the current global best set $G_{best}$. In other words, DMOPSO updates the $G_{set}$ set by finding the best solution for the $i^{th}$ subproblem from $\{P \cup G_{best}\}$ as it is shown in Algorithm 2. A more detailed description of DMOPSO can be found in [23].

# 3. OUR PROPOSED APPROACH FOR CONSTRAINED OPTIMIZATION

As indicated before, the proposed $\varepsilon$-constrained DMOPSO ($\varepsilon$DMOPSO) is based on the $\varepsilon$-constraint method [19] and it adopts the decomposition-based Multi-Objective Optimizer (DMOPSO), which was initially introduced for dealing with unconstrained MOPs [23]. Our proposed $\varepsilon$DMOPSO uses the information related to the whole swarm of particles in order to approximate solutions towards the Pareto optimal front while satisfying the constraints of the problem.

In the following description we assume that $W$ is a set of well-distributed weight vectors $\{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$. Along the search, $\varepsilon$DMOPSO finds the best solution to each subproblem defined by the Tchebycheff approach (equation (2)) using each weight vector in $W$. In this way, the objective function of the $j^{th}$ subproblem is then defined by $g^{te}(\mathbf{x}|\mathbf{w}^j, \mathbf{z})$, where $\mathbf{w}^j \in W$ and $\mathbf{z} = (z_1, \ldots, z_k)^T$ is the artificial utopian vector whose component $z_i$ is the minimum value found so far for the objective $f_i$.

At each generation, $\varepsilon$DMOPSO finds the best solution to each subproblem according to the decomposition approach and the proposed $\varepsilon$ level comparison. Throughout the search, $\varepsilon$DMOPSO maintains:

1. a population of $N$ particles $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, where $\mathbf{x}^i$ is the current solution to the $i^{th}$ subproblem;
2. $F^1, \ldots, F^N$, where $F^i$ is the $F$-value of $\mathbf{x}^i$, i.e., $F^i = \mathbf{F}(\mathbf{x}^i)$ for each $i = 1, \ldots, N$;

In an analogous way, $\varepsilon$DMOPSO adopts the same framework as DMOPSO (Algorithm 1). However, in order to satisfy the constraints of the problem, the updating algorithm (Algorithm 2) to select the global best set is modified. In the following section we present a detailed description of the proposed constraint-handling technique employed by $\varepsilon$DMOPSO.

## 3.1 Normalized Constraint Violation Degree

In order to measure an appropriate constraint violation between two solutions, we use the normalized constraint violation degree. In the following description, we assume that the constraint violation degree of a solution $\mathbf{x}$ is computed using equation (7).

Let $P$ be the set of particles found by $\varepsilon$DMOPSO. Assuming $p$ inequality constraints of the form $g_i \leq 0$ $(i = 1, \ldots, p)$, the normalized constraint violation of any solution $\mathbf{x}$ can be computed as:

$$\phi_N(\mathbf{x}) \quad = \quad \sum_{i=1}^{p} \left|\left| \max\left\{0, \frac{g_i(\mathbf{x})}{g_{\max}^i}\right\} \right|\right|^{\alpha} \tag{14}$$

where $\alpha$ is a positive number (in this work, $\alpha = 1$) and $g_{max}^i$ is the maximum violation value for the $i^{th}$ constraint in the set of solutions $P$. To be more precise, $g_{max}^i$ can be calculated by:

$$g_{\max}^i = \max_{\mathbf{x} \in P}\{0, g_i(\mathbf{x})\}, \quad i = 1, \ldots, p \tag{15}$$

Note however that if $g_{max}^i = 0$, all the solutions $\mathbf{x} \in P$ are feasible for the $i^{th}$ constraint and equation (14) cannot be computed. For this case, we set $g_{max}^i = 1$, thus penalizing the normalized constraint value $\phi_N(\mathbf{x})$ if solution $\mathbf{x}$ is not feasible for the $i^{th}$ constraint.

## 3.2 The $\varepsilon$ level value

For the $\varepsilon$-constraint method, we use the formulation that prefers a solution with strictly less function value than other one, i.e., the order relation $<_\varepsilon$ (equation (9)). In our approach, the $\varepsilon$ value is defined by:

$$\varepsilon = \phi_{N,min} + \tau \times (\phi_{N,max} - \phi_{N,min}) \tag{16}$$

where $\phi_{N,min}$ and $\phi_{N,max}$ are the minimum and the maximum value given by the normalized constraint violation of each solution $\mathbf{x}$ in $P$. To be more precise:

$$\begin{array}{rcl} \phi_{N,max} & = & \max_{\mathbf{x} \in P}\{\phi_N(\mathbf{x})\} \\ \phi_{N,min} & = & \min_{\mathbf{x} \in P}\{\phi_N(\mathbf{x})\} \end{array} \tag{17}$$

$\tau$ represents a parameter given by the user, which tries to balance the generation of feasible solutions with the minimization of each subproblem defined by $\varepsilon$DMOPSO.

The value of $\tau$ should be given in the range $[0, 1]$. If $\tau = 0$, the search is driven towards the generation of solutions with a normalized constraint value lower than $\phi_{N,min}$, i.e., preferring feasible solutions. If $\tau = 1$, the search is driven towards the generation of solutions with a lower cost for each subproblem, instead of preferring to satisfy the constraints of the CMOP.

## 3.3 Obtaining the global best set for constrained optimization

Along the cycles, each particle $\mathbf{x}^i$ in the swarm $P$ gives the best solution to the $i^{th}$ subproblem. Considering $N$ subproblems, the set $W$ has the same number of elements, i.e., the cardinality of $W$ is $N$. At the beginning, the set of particles $P$ constitutes the global best set $G_{best}$ (line 10 in Algorithm 1). Therefore, the number of solutions in $G_{best}$ is also $N$. The following updates of the global set are carried out in two different stages.

1. Preferring convergence. Let $T = \{P \cup G_{best}\}$ be the set of solutions to be considered as the new global best set $G_{best}$. We select from $T$, the set of solutions $C^1 = \{\mathbf{a}^1, \ldots, \mathbf{a}^N\}$ which minimize each subproblem defined by each $\mathbf{w}^i$ in $W$ by using Algorithm 2, i.e., $C^1 = update\_G_{best}(T, W)$.

   With this, we prefer solutions with a lower function value for each separate subproblem even if they do not satisfy the constraints of the CMOP.

2. Satisfying constraints. Let $R = \{T \setminus C^1\}$ be the remaining solutions in $T$. In an analogous way, we define a new set $C^2 = \{\mathbf{b}^1, \ldots, \mathbf{b}^N\}$ by using Algorithm 2, $C^2 = update\_G_{best}(R, W)$. It is worth noting that the selected solutions $\mathbf{a}^i$ and $\mathbf{b}^i$ correspond to the best solutions for the $i^{th}$ subproblem defined by $\mathbf{w}^i$.

   The tiebreaker for selecting a new solution as part of the new global best set $G_{best}$ is carried out by using the $\varepsilon$-constraint method according to the next rule.

   Let $g_1^{te}(g_2^{te})$ and $\phi_N^1(\phi_N^2)$ be the function values given by the Tchebycheff aggregation function (equation (2)) and the normalized constraint violation at the solution $\mathbf{a}^i(\mathbf{b}^i)$, respectively. Then, for any $\varepsilon$ satisfying $\varepsilon \geq 0$, the $\varepsilon$ level comparison $<_\varepsilon$ between $(g_1^{te}, \phi_N^1)$ and $(g_2^{te}, \phi_N^2)$ is defined as follows:

$$(g_1^{te}, \phi_N^1) <_\varepsilon (g_2^{te}, \phi_N^2) \Leftrightarrow \begin{cases} g_1^{te} < g_2^{te}, & \text{if } \phi_N^1, \phi_N^2 \leq \varepsilon \\ g_1^{te} < g_2^{te}, & \text{if } \phi_N^1 = \phi_N^2 \\ \phi_N^1 < \phi_N^2, & \text{otherwise} \end{cases} \quad (18)$$

   Therefore, $\mathbf{g}_{best}^i = \mathbf{a}^i$, if and only if equation (18) is satisfied. Otherwise $\mathbf{g}_{best}^i = \mathbf{b}^i$, for each $i \in \{1, \ldots, N\}$. The global best set for the next cycle of $\varepsilon$DMOPSO is then constituted by:

$$G_{best} = \{\mathbf{g}_{best}^i, \ldots, \mathbf{g}_{best}^N\}$$

With this, stage 1 finds convergence towards optimal values of each subproblem, while stage 2 promotes satisfying the constraints of the problem, i.e., by performing stage 2, the $\varepsilon$ level will reduce its value along the cycles trying to satisfy the constraints of the CMOP.

# 4. EXPERIMENTAL STUDY

## 4.1 Constrained Multi-Objective Airfoil Shape Optimization Problems

In order to assess the performance of our proposed approach we employ CMO-ASOPs adapted from [18, 22, 12, 2]. Two different CMO-ASOPs were defined having two and three objectives and different numbers of constraints. Since the test CMO-ASOPs deal with airfoil shape geometries, next we describe the geometry parameterization used in them, which in turn defines the dimensionality of the search space.

| Parameter | Lower bound | Upper bound |
|---|---|---|
| $r_{leup}$ | 0.0085 | 0.0126 |
| $r_{lelo}$ | 0.0020 | 0.0040 |
| $\alpha_{te}$ | 7.0 | 10.0 |
| $\beta_{te}$ | 10.0 | 14.0 |
| $Z_{te}$ | -0.0060 | -0.0030 |
| $\Delta Z_{te}$ | 0.0025 | 0.0050 |
| $X_{up}$ | 0.4100 | 0.4600 |
| $Z_{up}$ | 0.1100 | 0.1300 |
| $Z_{xxup}$ | -0.90 | -0.70 |
| $X_{lo}$ | 0.20 | 0.26 |
| $Z_{lo}$ | -0.0230 | -0.0150 |
| $Z_{xxlo}$ | 0.05 | 0.20 |

**Table 1: Parameter ranges for modified PARSEC airfoil representation**

### *4.1.1 Geometry parameterization*

Finding an optimum representation scheme for ASO problems is an important step for a successful aerodynamic optimization task.

In this case, the PARametric SECtion (PARSEC) airfoil representation [17] is used. Figure 1 illustrates the 11 basic parameters used for this representation. Where $r_{le}$ is the leading edge radius, $X_{up}$ is the location of maximum thickness for the upper surface (extrados), $X_{lo}$ is the location of maximum thickness for lower surface (intrados), $Z_{up}$ is the maximum thickness for upper surface, $Z_{lo}$ is the maximum thickness for lower surface, $Z_{xxup}$ is the curvature for upper surfaces, at maximum thickness location, $Z_{xxlo}$ is the curvature for lower surfaces, at maximum thickness location, $Z_{te}$ is the trailing edge coordinate, $\Delta Z_{te}$ is the trailing edge thickness, $\alpha_{te}$ is the trailing edge direction, and $\beta_{te}$ represents the trailing edge wedge angle.

For the present case, the modified PARSEC geometry representation adopted allows us to define independently the leading edge radius, both for upper and lower surfaces. Thus, 12 variables in total are used. Their allowable ranges are defined in Table 1.

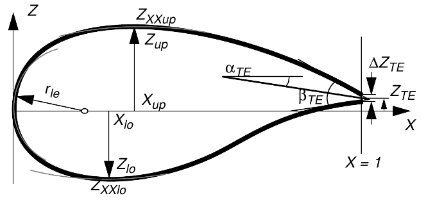

**Figure 1: PARSEC airfoil parameterization**

The PARSEC airfoil geometry representation uses a linear combination of shape functions for defining the upper and lower surfaces. These linear combinations are given by:

$$Z_{upper} = \sum_{n=1}^{6} a_n x^{\frac{n-1}{2}} \ , \ Z_{lower} = \sum_{n=1}^{6} b_n x^{\frac{n-1}{2}}$$

In the above equations, the coefficients $a_n$, and $b_n$ are determined as functions of the 12 described geometric parameters, by solving two systems of linear equations.

### 4.1.2 Definition of the CMO-ASOPs

The goal is to optimize the airfoil shape of a standard-class glider, aiming at obtaining optimum performance for this type of aircraft. In the following definition of CMO-ASOPs, common objectives are to maximize the aerodynamic efficiency, i.e., the lift to drag forces ratio $C_l/C_d$, to maximize the aerodynamic power efficiency, i.e., $C_l^3/C_d^2$, and to minimize the absolute value for the aerodynamic moment coefficient, $C_m$. These objectives reflect the global aerodynamic efficiency of a glider and are evaluated for different flight regimes, which are defined in terms of the airfoil's angle of attack ($\alpha$), and the gliding velocity, given by the Reynolds ($Re$) and Mach ($M$) numbers. Note that in the CMO-ASOPs definition, all objectives are considered to be minimized, requiring, in consequence, the use of the inverse of $C_l/C_d$ and $C_l^3/C_d^2$ ratios. For some airfoil designs, being evaluated, it can happen that its $C_m$ value could be positive or negative. In this case, we consider the moment coefficient value squared as the objective to minimize. In some other situations, existing designs serve as reference values for the new designs. In this case, constraints are imposed in order to have better performing designs. The mathematical definition for the CMO-ASOPs is presented next:

### *Test Problem* CMO-ASOP1

This problem considers two objective functions: *i)* to maximize the aerodynamic efficiency, i.e., the lift to drag forces ratio $C_l/C_d$, and *ii)* to minimize the absolute value for the aerodynamic moment coefficient. From the aerodynamic point of view, an airfoil's efficiency could be increased by reducing its thickness to chord ratio, $t/c$, leading to very thin airfoil shapes. In order to avoid this situation, one constraint is added to achieve a $t/c = 0.13$. This thickness ratio allows to reduce the wing weight while maintaining good structural strength. The mathematical definition for CMO-ASOP1 is:

i) $Minimize: C_d/C_l \text{ @ } \alpha = 4.0^o,\ Re = 2.0 \times 10^6,\ M = 0.1$

ii) $Minimize: C_m^2 \text{ @ } \alpha = 4.0^o,\ Re = 2.0 \times 10^6,\ M = 0.1$
s.t. $t/c = 0.13$

### *Test Problem* CMO-ASOP2

This problem considers three objective functions: *i)* to maximize the aerodynamic efficiency, i.e., to maximize the $C_l/C_d$ ratio at a first flight condition, *ii)* to maximize aerodynamic power efficiency, i.e., $C_l^3/C_d^2$ at a second flight, and *iii)* to maximize aerodynamic power efficiency, i.e., $C_l^3/C_d^2$ at a third flight condition. For this problem, four constraints are formulated. The first three aim to have a better aerodynamic objective function with respect to a reference airfoil shape commonly used in this application (the $a720$ airfoil) and reported in [18]. The fourth constraint aims to limit the $t/c$ ratio in a similar way as was done in the previous problem. The mathematical definition for CMO-ASOP2 is:

*i)* $Minimize: C_d/C_l \text{ @ } \alpha = 1.0^o,\ Re = 4.0 \times 10^6,\ M = 0.3$

*ii)* $Minimize: C_d^2/C_l^3 \text{ @ } \alpha = 3.0^o,\ Re = 3.0 \times 10^6,\ M = 0.3$

*iii)* $Minimize: C_d^2/C_l^3 \text{ @ } \alpha = 5.0^o,\ Re = 2.0 \times 10^6,\ M = 0.3$

$$\begin{array}{lll} \text{s.t.} & C_d/C_l & \leq 0.005201 \\ & C_d^2/C_l^3 & \leq 0.004462 \\ & C_d^2/C_l^3 & \leq 0.004458 \\ & t/c & = 0.13 \text{ for all flight operating conditions.} \end{array}$$

## 4.2 Performance Measures

### *Hypervolume*

The Hypervolume ($I_H$) performance measure was proposed in [26]. This performance measure is Pareto compliant [27], and quantifies both convergence and spread of nondominated solutions along the Pareto optimal front. It is mathematically stated as:

$$I_H(P) = \Lambda\left(\bigcup_{\mathbf{p}\in P}\{\mathbf{x}|\mathbf{p} \prec \mathbf{x} \prec \mathbf{r}\}\right) \tag{19}$$

where $\Lambda$ denotes the Lebesgue measure, $P$ is the Pareto approximation to the true Pareto front and $\mathbf{r} \in \mathbb{R}^k$ denotes a reference vector being dominated by all valid candidate solutions in $P$. A high $I_H$ value, indicates that the approximation $P$ is close to $PF$ and has a good spread towards the extreme portions of the Pareto front.

### *Feasibility Ratio*

The feasibility ratio ($I_F$) indicator refers to the ratio of the number of feasible solutions found in the final approximation $P$ to the Pareto front. It is mathematically stated as:

$$I_F(P) = \frac{P_f}{|P|} \tag{20}$$

where $P_f$ denotes the number of feasible solutions in $P$ and $|P|$ represents the cardinality of the population $P$.

## 4.3 Experimental Setup

To assess the performance over the CMO-ASOPs adopted, we compared the results obtained by our $\varepsilon$DMOPSO with respect to those achieved by the proposal presented by Jan and Zhang in [9], which we call here cMOEA/D-DE. We selected cMOEA/D-DE for its ability to solve complex problems [9] and because it uses one of the most representative MOEAs based on decomposition, the well-known MOEA/D-DE. Additionally, its source code is in the public domain.

In order to allow a fair comparison, the set of weight vectors was the same for both algorithms, and they were generated in the same way using the procedure described in [25], i.e., the settings of $N$ and $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ were controlled by a parameter $H$. More precisely, $\mathbf{w}^1, \ldots, \mathbf{w}^N$ are all the weight vectors in which each individual weight $w_j^i$ ($i = 1, \ldots, N$ and $j = 1, \ldots, k$) takes a value from: $\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\}$. Therefore, the number of such vectors in $W$ is given by $N = C_{H+k-1}^{k-1}$, where $k$ is the number of objective functions.

Because of the high computational time required by the CMO-ASOPs adopted in our study, we used a low number of subproblems. Here, we use $H = 49$ (for the bi-objective problem) and $H = 13$ (for the three-objectives problem), i.e., we generated 50 and 105 weight vectors for each CMO-ASOP, respectively. It is worth noting that the use of large populations (greater than 200 individuals), could significantly increase the computational time (even requir-

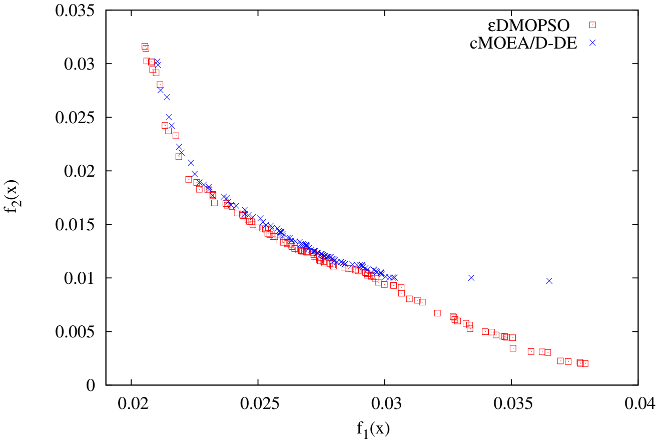

**Figure 2: $\varepsilon$DMOPSO and MOEA/D-DE for CMO-ASOP1.**

ing days in some cases) that an optimizer requires (only for a single run) for solving these CMO-ASOPs.

For each CMO-ASOP, we performed 30 independent runs with each algorithm. The parameters for both algorithms where set as follows: $N = 50$ and $N = 105$ represent the number of weight vectors (and the population size) for CMO-ASOP1 and CMO-ASOP2, respectively. The maximum number of iterations was set as $N_{it} = 100$ for both test problems. Therefore, both algorithms performed 5,000 and 10,500 fitness function evaluations for each problem, respectively. For cMOEA/D-DE, the parameters were set as: $T_n = \lceil 0.1 \times N \rceil$ (neighborhood size) $\delta = 0.9$ (probability that parent solutions are selected from the neighborhood) $F = 0.5$ (the differential factor), $CR = 1$ (the crossover ratio), $\eta_m = 20.0$ (the mutation index), $P_m = 1/n$ (mutation rate) and $n_r \lceil 0.01 \times N \rceil$ (the number of solutions to be replaced). These values were set as it is suggested in [9]. For the constraint-handling technique, the parameters for cMOEA/D-DE we used were: $s_1 = 0.01$ and $s_2 = 20.0$, and they were set as in [9] (we used the best parameters settings found by Jan and Zhang in their comparative study).

For $\varepsilon$DMOPSO, $\tau = 0.3$ and $T_a = 2$ represent the control parameter for equation (18) and the age threshold, respectively. The constraints on the velocity $(c_1, c_2)$ and the inertia factor $(w)$ were dynamically defined. As in DMOPSO [23], we adopted uniformly distributed values, such that: $c_1, c_2 \in (1.2, 2.0)$ and $w \in (0.1, 0.5)$.

## 5. DISCUSSION OF RESULTS

As indicated before, the results obtained by our proposed approach (i.e., $\varepsilon$DMOPSO) were compared against those produced by cMOEA/D [9]. In Table 2, the hypervolume values were calculated using the reference vectors $\mathbf{r}$ given in this same table. According to the results presented in Table 2, $\varepsilon$DMOPSO had a better performance than cMOEA/D-DE in the two CMO-ASPs adopted. This table provides a quantitative assessment of the performance of $\varepsilon$DMOPSO in terms of the $I_H$ and $I_F$ indicators. That means that the solutions obtained by $\varepsilon$DMOPSO achieved a better approximation of the Pareto optimal front than the solutions obtained by cMOEA/D-DE while maintaining an acceptable ratio of feasible solutions. Due to space limitations, we only show (in Figure 2) the performance of $\varepsilon$DMOPSO when it is compared with respect to cMOEA/D-DE for CMO-ASOP1. As we can see, $\varepsilon$DMOPSO produced solutions with better convergence to the extremes and most of them dominated those produced by cMOEA/D. This corroborates the results of the good performance of our proposed $\varepsilon$DMOPSO regarding CMO-ASOP1. For CMO-ASOP2, numerical results showed that our proposed approach significantly outperformed cMOEA/D. In fact, a good value of $I_H$ indicates that $\varepsilon$DMOPSO achieved a better convergence towards the Pareto optimal front[2] than cMOEA/D.

In terms of the feasibility of the solutions, we can see from Table 2 that our proposed $\varepsilon$DMOPSO obtained a better ratio of feasible solutions for both problems (CMO-ASOP1 and CMO-ASOP2). This means that our proposal of incorporating the $\varepsilon$-constraint method into DMOPSO, seems, at first sight, a good strategy to deal with expensive optimization problems. Although our proposed approach has been tested here only with CMO-ASOPs, it can clearly be applied to other real-world problems as well, and that is part of our ongoing research.

## 6. CONCLUSIONS AND FUTURE WORK

We have proposed a new approach based on DMOPSO for constrained multi-objective optimization. Our proposed $\varepsilon$DMOPSO introduces a selection mechanism based on the $\varepsilon$-constraint method in order to define the set of global best solutions along the search process. Although the use of the new constraint-handling technique was adopted here for DMOPSO, this approach can be easily adopted by any other decomposition-based MOEA which uses a $(\mu + \lambda)$-selection mechanism to obtain optimal solutions (for example, the approach presented in [1]). Our preliminary results indicate that our proposed $\varepsilon$DMOPSO is highly competitive with respect to a current state-of-the-art MOEA (i.e., cMOEA/D-DE) in the test problems adopted here. However, more test problems are required in order to draw more general conclusions about its performance.

As part of our future work, we would like to validate our proposed approach in many-objective optimization problems (i.e., multi-objective problems having more than 3 objectives). We also intend to focus on the design of a strategy that allows us to adjust, in a dynamic way, the value of $\varepsilon$ employed by our proposed $\varepsilon$DMOPSO. We are also considering the possibility of experimenting with a penalty function or another powerful constraint-handling technique such as stochastic ranking [16]. Furthermore, in order to deal in an efficient way with more complex CMOPs, the introduction of local search mechanisms to $\varepsilon$DMOPSO seems as a promising path for future research. We believe that the use of an appropriate local search mechanism coupled with a MOEA (such as those presented in [10, 24]) could give rise to a powerful search engine capable of dealing with more complex CMOPs in a more effective and efficient way.

## Acknowledgements

The second author acknowledges support from IPN-ESIME through the COTEBAL, EDI and SIBE programs.

[2] For CMO-ASOP1 and CMO-ASOP2, the true Pareto fronts are unknown.

Table 2: Results of $I_H$ for $\varepsilon$DMOPSO and cMOEA/D-DE

| MOP | $\varepsilon$DMOPSO | | cMOEA/D-DE | | reference vector **r** |
|---|---|---|---|---|---|
| | $I_H$ | $I_F$ | $I_H$ | $I_F$ | |
| | *average*<br>($\sigma$) | *average*<br>($\sigma$) | *average*<br>($\sigma$) | *average*<br>($\sigma$) | |
| CMO-ASOP1 | **9.3477e-04**<br>(1.0504e-05) | **81.37**<br>(0.093454) | 8.6921e-04<br>(1.5823e-05) | 76.72<br>(0.021125) | $(0.065, 0.031)^T$ |
| CMO-ASOP2 | **1.5333e-12**<br>(6.2821e-13) | **67.20**<br>(0.012837) | 9.0066e-16<br>(4.8502e-15) | 44.72<br>(0.982360) | $(0.0053, 0.0045, 0.0045)^T$ |

# 7. REFERENCES

[1] A. Arias Montaño, C. A. Coello Coello, and E. Mezura-Montes. MODE-LD+SS: A Novel Differential Evolution Algorithm Incorporating Local Dominance and Scalar Selection Mechanisms for Multi-Objective Optimization. In *2010 IEEE Congress on Evolutionary Computation (CEC'2010)*, pages 3284–3291, Barcelona, Spain, July 18–23 2010. IEEE Press.

[2] A. Arias-Montaño, C. A. Coello Coello, and E. Mezura-Montes. Multi-Objective Airfoil Shape Optimization Using a Multiple-Surrogate Approach. In *CEC'2012*, pages 1188–1195, Brisbane, Australia, June 10-15 2012. IEEE Press.

[3] A. Arias-Montaño, C. A. Coello Coello, and E. Mezura-Montes. Multi-Objective Evolutionary Algorithms in Aeronautical and Aerospace Engineering. *IEEE Transactions on Evolutionary Computation*, 16(5):662–694, October 2012.

[4] M. Clerc. Confinements and Biases in Particle Swarm Optimisation, 2006.

[5] C. A. Coello Coello. Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer methods in applied mechanics and engineering*, 191:1245–1287, 2002.

[6] C. A. Coello Coello and G. B. Lamont, editors. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, 2004. ISBN 981-256-106-4.

[7] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[8] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition, June 2005.

[9] M. A. Jan and Q. Zhang. Moea/d for constrained multiobjective optimization: Some preliminary experimental results. In *2010 UK Workshop on Computational Intelligence (UKCI)*, pages 1–6, 2010.

[10] P. Koch, O. Kramer, G. Rudolph, and N. Beume. On the hybridization of sms-emoa and local search for continuous multiobjective optimization. In *GECCO'2009*, pages 603–610, New York, NY, USA, 2009. ACM.

[11] W. F. Leong. *Multiobjective particle swarm optimization: Integration of dynamic population and multiple-swarm concepts and constraint handling*. PhD thesis, Oklahoma State University, 2008.

[12] A. López-Jaimes, A. Arias-Montaño, and C. A. Coello Coello. Preference Incorporation to Solve Many-Objective Airfoil Design Problems. In *CEC'2011*, pages 1605–1612, New Orleans, Louisiana, USA, 5-8 June 2011. IEEE Service Center.

[13] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachuisetts, 1999.

[14] N. A. Moubayed, A. Petrovski, and J. A. W. McCall. A novel smart multi-objective particle swarm optimisation using decomposition. In *PPSN XI*, volume 6239, pages 1–10, 2010.

[15] M. Reyes-Sierra and C. A. Coello Coello. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.

[16] T. P. Runarsson and X. Yao. Stochastic Ranking for Constrained Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 4(3):284–294, September 2000.

[17] H. Sobieczky. Parametric airfoils and wings. In K. Fuji and G. S. Dulikravich, editors, *Notes on Numerical Fluid Mechanics, Vol.. 68*, pages 71–88, Wiesbaden, 1998. Vieweg Verlag.

[18] A. Szöllös, M. Smid, and J. Hajek. Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. *Advances in engineering software*, 40(6):419–430, June 2009.

[19] T. Takahama, S. Sakai, and N. Iwane. Constrained Optimization by the $\varepsilon$ Constrained Hybrid Algorithm of Particle Swarm Optimization and Genetic Algorithm. In S. Zhang and R. Jarvis, editors, *AI 2005: Advances in Artificial Intelligence*, volume 3809 of *Lecture Notes in Computer Science*, pages 389–400. Springer Berlin Heidelberg, 2005.

[20] Y.-y. Wang, B.-q. Zhang, and Y.-c. Chen. Robust airfoil optimization based on improved particle swarm optimization method. *Applied Mathematics and Mechanics*, 32(10):1245–1254, 2011.

[21] U. K. Wickramasinghe, R. Carrese, and X. Li. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *CEC'2010*, pages 1857–1869, Barcelona, Spain, July 18–23 2010. IEEE Press.

[22] S. Zapotecas Martínez, A. Arias Montaño, and C. A. Coello Coello. A Nonlinear Simplex Search Approach for Multi-Objective Optimization. In *2011 IEEE Congress on Evolutionary Computation (CEC'2011)*, pages 2367–2374, New Orleans, Louisiana, USA, 5–8 June 2011. IEEE Service Center.

[23] S. Zapotecas Martínez and C. A. Coello Coello. A Multi-objective Particle Swarm Optimizer Based on Decomposition. In *GECCO'2011*, pages 69–76, Dublin, Ireland, July 2011. ACM Press.

[24] S. Zapotecas Martínez and C. A. Coello Coello. A Direct Local Search Mechanism for Decomposition-based Multi-Objective Evolutionary Algorithms. In *CEC'2012*, pages 3431–3438, Brisbane, Australia, June 2012. IEEE Press.

[25] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731,| December 2007.

[26] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *PPSN V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[27] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.