Proceedings of ASME Turbo Expo 2011
GT2011
June 6-10, 2011, Vancouver, British Columbia, Canada

GT2011-45069

EFFICIENT GRADIENT-BASED ALGORITHMS FOR THE CONSTRUCTION OF PARETO FRONTS

Sriram Shankaran*and Brian Barr
General Electric - Global Research Center, Niskayuna, NY

ABSTRACT

The objective of this study is to develop and assess a gradient-based algorithm that efficiently traverses the Pareto front for multi-objective problems. We use high-fidelity, computationally intensive simulation tools (for eg: Computational Fluid Dynamics (CFD) and Finite Element (FE) structural analysis) for function and gradient evaluations. The use of evolutionary algorithms with these high-fidelity simulation tools results in prohibitive computational costs. Hence, in this study we use an alternate gradient-based approach. We first outline an algorithm that can be proven to recover Pareto fronts. The performance of this algorithm is then tested on three academic problems: a convex front with uniform spacing of Pareto points, a convex front with non-uniform spacing and a concave front. The algorithm is shown to be able to retrieve the Pareto front in all three cases hence overcoming a common deficiency in gradient-based methods that use the idea of scalarization. Then the algorithm is applied to a practical problem in concurrent design for aerodynamic and structural performance of an axial turbine blade. For this problem, with 5 design variables, and for 10 points to approximate the front, the computational cost of the gradient-based method was roughly the same as that of a method that builds the front from a sampling approach. However, as the sampling approach involves building a surrogate model to identify the Pareto front, there is the possibility that validation of this predicted front with CFD and FE analysis results in a different location of the "Pareto" points. This can be avoided with the gradient-based method. Additionally, as the number of design variables increases and/or the number of required points on the Pareto front is reduced, the computational cost favors the gradient-based approach.

*Corresponding author: shankaran@ge.com

1 Nomenclature

$\mathbf{x}$	Vector of Design Variables
$\mathbf{f}(\mathbf{x})$	Vector of Objective Functions
g	Vector of inequality constraints
$\mathbf{x}^*$	Pareto Optimal decision variables
$\mathscr{F}$	Feasible region of the design space
$\mathbf{u},\mathbf{v}$	Two possible optimal solutions
$\mathscr{P}^*$	Pareto Optimal Set
$\mathscr{P}\mathscr{F}^*$	Pareto Front
A	Gradient matrix
h	Objective Function for the auxiliary problem
v	Search direction for the auxiliary problem
$\mathscr{J}\mathbf{f}$	Gradient-matrix
t	A scalar that represents the step-size
ε	A small real number to represent tolerance
$\mathbf{x}_{\text{base}}$	Baseline and starting point for Pareto Algorithm

2 Introduction

It is common during the engineering design process of an overall system to consider trade-offs between various performance metrics. Typically, these trade-offs occur across various sub-systems like aerodynamics, structures, heat transfer, controls etc. It is also possible that these trade-offs occur within a particular sub-system (for example, higher lift on turbine blades typically occur at the cost of increased profile drag). Knowledge of the cost of trade-offs is important to a systems-level designer as it ensures that the overall system metrics are satisfied while maintaining "equilibrium" among the constituents.

There exists different notions of equilibrium solutions, and each of them provide a different trade-off between the competing metrics. The most common among them are Nash equilibrium solutions for non-cooperative sub-systems and Pareto optimal solutions for cooperative sub-systems. Most engineering systems can be viewed as composed of cooperative sub-systems with complete knowledge of performance metrics, constraints and design variables of all other sub-systems. Nash equilibrium solutions find use in design processes that account for competition. Typically, Nash solutions are non-unique, and more importantly not stable to small perturbations in the design variables. It is also, in general, harder to obtain Nash solutions for complex engineering systems.

As our focus is on common engineering systems we explore Pareto solutions. There exists many techniques to recover Pareto optimal solutions. As the concept of Pareto is rooted the non-dominance of a solution by other solutions, the process of finding a Pareto solution is essentially one of ordering. If one could generate an ensemble of solutions, the Pareto ordering can be used to filter the dominated from the non-dominated solutions. The Pareto ordering process and its use with population-based mutation operators is typically used in evolutionary approaches to determine Pareto solutions and the Pareto front. This approach is straight-forward but requires knowledge or construction of a model for the sub-systems. It can handle convex and concave Pareto fronts and furthermore, discontinuous/disjointed Pareto fronts. The latter is useful if the active constraints force certain portions of the front to the infeasible. When the sub-systems are governed by Partial Differential Equations (PDEs) as with fluid systems, structural mechanics and heat transfer, the construction of an approximate surrogate model is necessary as evaluation of the sub-system performance with the numerical solvers for the PDEs is expensive. Assuming that an accurate lower-fidelity model can be determined, this approach requires verification of Pareto front and solutions using the more high-fidelity PDE solver. It is possible that the trade-offs suggested from using the lower order model are not always accurate. Hence, it is beneficial if the PDE solver can be used during determination of the Pareto front and Pareto solutions.

There have been a wide range of studies that use evolutionary algorithms to determine the Pareto front [1–3] (and references therein). As an alternative to the evolutionary algorithm approach to determining the Pareto front, one could use a gradient-based approach [4, 5]. In theory the gradient-based approach can be used with the surrogate model but it does not address the issue of possible “inconsistency” between the solutions obtained with the high and low fidelity models. Gradient-based algorithms for Multi Objective Problems (MOP) are typically based on a scalarization concept where the objectives are combined to form a single objective function [6, 7]. Hence, iso-contours of this single objective functions are straight lines (hyperplanes in higher dimensional spaces) in objective function space. The slope and intercept of the straight-lines (or hyperplanes) depend on the choice of scalars used to combine the objectives [8]. Hence, by altering the slopes and intercept of the hyperplanes different points on the convex front can be obtained. This approach has many well-known defects, namely the inability to capture concave portions and disjoint Pareto fronts. In addition, an evenly distributed choice of scalar weights between $[0,1]$ need not generate equi-spaced points on the Pareto front. Typically, the scalarization approach needs to be supplemented with a technique to dynamically alter the weights [9]. Apart from the end points, concave fronts are typically not recovered. Hence, additional heuristic/algorithms need to be devised to capture the concave portions. To overcome these difficulties an alternate approach for gradient-based algorithms was outlined in [10]. In a series of articles [11] and a U.S. patent [12], a technique similar to what is described here and in [10] was used to show the possibility of gradient-based algorithms for simultaneous improvement of multiple objectives and tracking of Pareto front. The method in [11] has been further enhanced with genetic algorithms to result in a hybrid algorithm that is reported to have superior abilities to capture complicated Pareto fronts.

The layout of the paper is as follows. In Section 3, the basic definitions of Pareto solutions are described. Here, we also discuss the mathematical construct of scalarization, the relation of the weights to the hyperplanes (that are tangential to the front) and show that the scalarization approach captures the convex portions of the Pareto front. In Section 4, following [10], we outline an algorithm to capture the Pareto front by traversing from the minima of a single-objective optimization problem. In Sections 5 and 6 we analyze this algorithm to determine how it captures the different portions of convex (uniform and non-uniform) and concave Pareto fronts. In Section 7, we use this algorithm for an multi-disciplinary problem in turbo-machinery where a low fidelity Navier-Stokes CFD solver analysis tool (MISES [13]) is used for aerodynamic performance predictions and ANSYS [14] is used to determine the modal response for an aero-structural optimization problem. All gradients are estimated with finite-difference gradients by using the above analysis tools. Finally, we present our conclusions in Section 8.

3 Preliminaries

Pareto optimal solutions are commonly used in Engineering studies to study trade-offs between various competing performance metrics. Pareto solutions fall under the category of Multi-Objective Optimization Problems (MOOP) which is also called vector optimization problems and can be notionally defined as

$$\begin{aligned} \min\ & \mathbf{f}(\mathbf{x}) := (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_s(\mathbf{x})) \\ \text{s.t.}\ & g_i(\mathbf{x}) \leq 0, \quad i = 1, 2, ..., m \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is a vector of control variables (possible bounded), $f_i : \mathbb{R}^n \to \mathbb{R},\ \ i = 1,2,...,s$ are the objective functions and $g_i : \mathbb{R}^n \to \mathbb{R}\ \ \forall i = 1,2,...,m$ are the constraints which we will assume them to be continuously differentiable.

There exists a variety of solution concepts for these multi-objective problems and Pareto optimality and Nash equilibrium solutions have been widely studied. The former is used in co-operative systems and the latter is more common when studying non-cooperative scenarios. Pareto optimal solutions can be defined using the following notions.

Definition 3.1. *We say that a vector of decision variables $\mathbf{x}^* \in \mathscr{F}$ is Pareto optimal if there does not exist another $\mathbf{x} \in \mathscr{F}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*) \forall i \in (1,2,...,s)$ and $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at-least one j. Here, $\mathscr{F}$ denotes the feasible region of the problem (i.e., where the constraints are satisfied).*

In words, this definition says that $\mathbf{x}^*$ is Pareto optimal if there exists no feasible vector of decision variables $\mathbf{x} \in \mathscr{F}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion. Unfortunately, this concept almost always gives not a single solution, but rather a set of solutions called the Pareto optimal set. The vectors $\mathbf{x}^*$ corresponding to the solutions included in the Pareto optimal set are called non-dominated. The plot of the objective functions whose non-dominated vectors are in the Pareto optimal set is called the Pareto front.

Definition 3.2. *A vector $\mathbf{u}$ is said to dominate a vector $\mathbf{v}$ (denoted by $\mathbf{u} \preceq \mathbf{v}$) if and only if $\mathbf{u}$ is partially less than $\mathbf{v}$ i.e. $u_i \leq v_i \forall i \in (1,2,...,s)\ \wedge \exists i \in (1,2,...,s) : u_i < v_i$.*

Definition 3.3. *A Pareto optimal set $\mathscr{P}^*$ is defined as*

$$\mathscr{P}^* := \{\mathbf{x} \in \mathscr{F} | \neg \exists \mathbf{x}' \in \mathscr{F} \quad s.t.\ \ \mathbf{f}(\mathbf{x}') \preceq \mathbf{f}(\mathbf{x})\} \tag{2}$$

and this defines the Pareto front as

$$\mathscr{P}\mathscr{F}^* := \{\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_s(\mathbf{x})) | x \in \mathscr{P}^*\}$$

Following [10], the above definition of Pareto optimal points are now cast in terms of the gradient matrix, $\mathscr{J}\mathbf{f}$. We consider the unconstrained problem in Equation 2 for simplicity (Please refer to [10] for extensions to the constrained case). Assume that the design space, $\mathbf{x} \in \mathbb{R}^n$, hence $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^s$. The gradient matrix, $\mathscr{J}\mathbf{f}$, is a $s \times n$ matrix with entries $(\mathscr{J}\mathbf{f})_{ij} = \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j}(\mathbf{x})$. If we denote the set of strictly positive real numbers as $\mathbb{R}_{++}$, a point, $\mathbf{x}$ is locally Pareto optimal if

$$\text{range}(\mathscr{J}\mathbf{f}) \ \bigcap \ (-\mathbb{R}_{++})^s = \emptyset \tag{3}$$

and call these points Pareto critical. Hence, if a point is Pareto critical, then the condition in Equation 3 will only be satisfied by a direction which is identically zero. If a point is not Pareto critical, then there exists a direction $v \in \mathbb{R}^n$ satisfying

$$\text{range}(\mathscr{J}\mathbf{f}) \in (-\mathbb{R}_{++})^s$$

which is a descent direction for the vector-valued objective function $\mathbf{f}$. So in general, if a point $\mathbf{x}$ does not satisfy Equation 3, then one can compute a descent direction v with a suitable step-size length to form a new point.

3.1 Search Direction

Define A as the gradient matrix, $\mathscr{J}\mathbf{f}$ and the function $h : \mathbb{R}^n \to \mathbb{R}^s$ by

$$h(v) := \max\ \ (Av)_i, i = 1,2,...,s$$

h is convex (as it is maximum of the linear functions) and positive homogeneous. Consider the unconstrained minimization problem that finds a v to solve the following problem:

$$\begin{aligned} \min \quad & h(v) \\ \text{s.t. } & ||v|| \leq 1 \end{aligned} \tag{4}$$

Since the objective function in the above problem is proper, closed and strongly convex, it has a unique solution.

Lemma 3.4. *Let $v(\mathbf{x})$ be the solution of the optimal value problem in Equation 4.*

1. *If $\mathbf{x}$ is Pareto critical, then $v(\mathbf{x}) = 0$*
2. *If $\mathbf{x}$ is not Pareto critical, then*

$$(\mathscr{J}\mathbf{f}(v))_i \leq h(v) < 0,\ \ i = 1,2,...,s$$

3. *The mapping $\mathbf{x} \mapsto v(x)$ is continuous*

Proof. See [10].

There are different possible auxiliary problems that can be solved to determine the search direction. The one here leads to a simple linear problem. Irrespective of the form of the auxiliary problem, it is not required to solve them exactly.

3.2 Step-size Computation

Starting with a search direction, v, where $\mathscr{J}\mathbf{f}(v) < 0$, we choose an equal partition of the variable, t in $(0,1]$ and evaluate $\mathbf{f}(\mathbf{x}+tv)$ for all values of t. Among the evaluated values, we choose the value of t (and hence $\mathbf{x}$ that results in the smallest value of $\mathbf{f}$. Note that when we compare vectors we say that $\mathbf{f}^1 \leq \mathbf{f}^2$ in a component-wise sense. At the Pareto critical point, v is identically zero which will terminate the descent to the Pareto front. Please refer to [10] for existence results of an $t = \varepsilon > 0$ that satisfies the requirement that the step along the descent direction leads to an improvement of the MOP problem. Alternate forms of the line-search algorithm, like bisection can also be used but the equal partition method enables use of coarse-grain parallel computing to accelerate the overall algorithm.

4 Algorithms

The notions of Pareto-optimal sets and the Pareto front are easy to understand when surrogate models for the objective functions are employed. In particular, one constructs an "accurate" model from a few evaluations and uses repeated evaluations of this simplified model to sort and identify non-dominated solutions leading to the Pareto-optimal set and front. The evaluations of the model are coupled with evolutionary methods to enable construction of concave and disjointed portions of the Pareto fronts. This last feature makes them more attractive than the use of gradient-based techniques that traditionally have difficulty recovering such fronts.

Gradient-based algorithms use a combination of two mathematical constructs to determine the front. The first involves the scalarization of the MOOP to a single objective optimization problem (SOOP) using positive weights, namely, rewriting the MOOP as a standard optimization problem of the form:

$$\begin{aligned} \min \quad & F(\mathbf{x}) : F(\mathbf{x}) = \sum_i w_i f_i(\mathbf{x}) \\ \text{s.t.} \quad & \sum_i w_i = 1 \end{aligned} \tag{5}$$

The second involves the mathematical condition satisfied on the Pareto front that enables traversing of the front starting from one Pareto optimal point.

A straight-forward way of recovering the Pareto front is to start pose a set of optimization problems for differing values of the weights from a given starting location. Apart from the inherent computational cost, this process does not guarantee sufficient coverage of Pareto front as Pareto optimal points can be clustered around weights that differ slightly and could lead to the same points (as positiveness of the weights does not guarantee uniqueness of the Pareto optimal points). In any case, the possible combinations of weights increases exponentially with increasing number of objectives. As an alternate approach, we use the properties of the Pareto front to guide a gradient-based descent process that starts at one corner of the Pareto front and traces its way around the front. Issues related to concavity and disjoint fronts will be addressed later.

For ease of exposition, we focus on bi-objective problems. Introducing a real constant, $0 \leq \lambda \leq 1$, we can express the necessary condition for the Pareto optimality of a point as

$$\lambda \nabla f_1(\mathbf{x}^*) + (1-\lambda)\nabla f_2(\mathbf{x}^*) = 0 \tag{6}$$

which can be expressed for some $r \geq 0$ as $\nabla f_1(\mathbf{x}^*) = -r\nabla f_2(\mathbf{x}^*)$. Hence, for Pareto optimal points the gradients with respect to the objective functions point are parallel and point in opposite directions. As $\lambda = 0$ is also a Pareto optimal point, we can trace the Pareto front by starting at this point (the task of identifying this point is a single objective optimization problem) and follow the front in the direction of $\nabla f_2(\mathbf{x}^*)$. While $\nabla f_2(\mathbf{x}^*)$ provides the direction of movement, the distance to travel is obtained from a line-search like algorithm where the largest distance at which the condition

$$\nabla f_1(\mathbf{x}^*) = -r\nabla f_2(\mathbf{x}^*) \tag{7}$$

is satisfied is taken as the next point on the front. Note that this approach does not ensure recovery of equi-spaced points along the front but this can be achieved by adding λ as an additional minimizing variable (in addition to $\mathbf{x}$) and the added constraint that $\mathbf{f}(\mathbf{x}) - \mathbf{f}_{prev} = \gamma$ where γ is the desired spacing between Pareto optimal points [15].

The use of the scalarization approach and Equation 6 allows traversal of the front. In general, the step taken from the Pareto front (Equation 7 may not ensure that the new point lies on the front. Furthermore, for problems with more than 2 objectives, the necessary condition for Pareto optimality requires the condition of parallel gradient vectors for various combinations of more than 1 scalar. To overcome these problems, for general problems, it is useful to take a step in the direction of $\nabla f_2(\mathbf{x}^*)$ and solve the auxiliary problem in Equation 4 with an appropriate step-size.

The auxiliary problem does not require explicit solution. Once the gradient matrix, $\mathscr{J}\mathbf{f}$ is assembled, a sampling procedure can be used to determine the direction of improvement. Under the condition that $||v|| = 1$, the search direction is restricted to a hypersphere of dimension n. On this hypersphere, we are interested in vectors that provide simultaneous improvements in all objectives. For a bi-objective problem, we are interested in the portion of the two dimensional space of $\nabla f_{1_v}(\mathbf{x})$ and $\nabla f_{2_v}(\mathbf{x})$

that lies in the $(-,-)$ quadrant. Hence, a simple sampling strategy that polls the hypersphere and picks the search direction that satisfies Equation 4 is used to update the design variables.

The basic algorithm is summarized in Algorithm 1. Note that in the above algorithm, there are two gradient evaluations (assuming one step in the auxiliary problem) to move from Pareto critical point to the next. This will be expensive for a problem with a large n if the gradient evaluation algorithm uses the finite-differences. Obviously this can be circumvented with adjoint-based methods but may not always be available. To mitigate this cost, one can eliminate one of the gradient calculations by perturbing a Pareto critical point in an arbitrary direction and using the auxiliary problem to return back to the Pareto front. It is possible that the choice of the perturbation direction can have a significant effect on the coverage of the front and even the ability of the auxiliary problem to converge. A schematic of the overall process is shown in Figure 1.

Data: Set initial point, $\mathbf{x}$ to $\mathbf{x}_{\text{base}}$
while $\|\nabla f_1(\mathbf{x})\| \neq 0$ **do**
  Compute $\nabla f_1(\mathbf{x})$; $\mathbf{x}^{\mathbf{new}} = \mathbf{x} - \sigma \nabla f_1(\mathbf{x})$, σ is a constant;
  $\mathbf{x} = \mathbf{x}^{\mathbf{new}}$;
end
$\mathbf{x}^{\mathbf{end1}} = \mathbf{x}$;
while $\|\nabla f_2(\mathbf{x})\| \neq 0$ **do**
  Compute $\nabla f_2(\mathbf{x})$; $\mathbf{x}^{\mathbf{new}} = \mathbf{x} - \sigma \nabla f_2(\mathbf{x})$, σ is a constant;
  $\mathbf{x} = \mathbf{x}^{\mathbf{new}}$;
end
$\mathbf{x}^{\mathbf{end2}} = \mathbf{x}$;
$\mathbf{x} = \mathbf{x}^{\mathbf{end1}}$;
while $\|\mathbf{x} - \mathbf{x}^{\mathbf{end2}}\| \neq 0$ **do**
  Compute $\nabla f_2(\mathbf{x})$; Perturb $\mathbf{x}$ using $\mathbf{x}' = \mathbf{x} - \sigma \nabla \mathbf{f_2}(\mathbf{x})$, σ is a constant;
  while $v(\mathbf{x}) \neq 0$ **do**
    Compute $\mathscr{J}\mathbf{f}$ for $\mathbf{x}'$;
    Solve the auxiliary problem Equation 4 for v;
    Use v to compute t, that minimizes $\mathbf{f}$;
  end
  Update $\mathbf{x}^{\mathbf{new}} = \mathbf{x}' + \mathbf{t}\ \mathbf{v}(\mathbf{x})$;
  $\mathbf{x} = \mathbf{x}^{\mathbf{new}}$;
end

Algorithm 1: A Steepest-Descent version for MOOP.

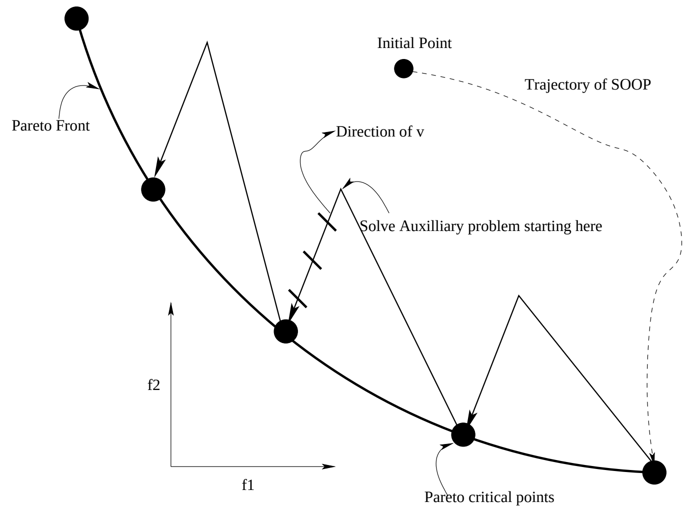


Figure 1. A schematic outlining the trace of the algorithm during the Pareto front tracking process

4.1 A simple one-dimensional problem to illustrate the algorithm

We now expose some of the details of the algorithm by considering a simple problem with two objectives with quadratic form in one design variable. $f_i = (x - c_i)^2$, $c_i = 0, 2 \;\; i = 1, 2$. Hence, f_1 is "centered" at the origin and f_2 is "centered" at 2. The Pareto front in objective space (f_1, f_2) is a convex curve obtained by varying the design variable $\mathbf{x}$ in $[0, 2]$.

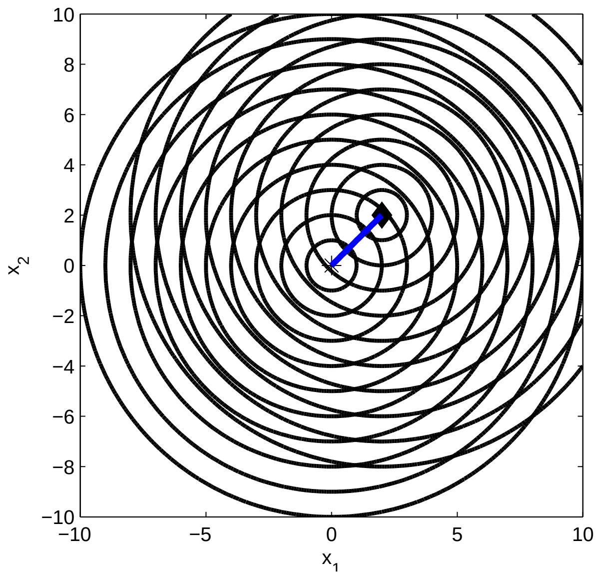
When $\mathbf{x} \in (0, 2)$, Equation 7 is satisfied, hence we would expect that v is identically zero. Consider the gradient matrix, $\mathscr{J}\mathbf{f}$ in the direction of v (v varies in $[0, 1]$). This has two entries $2xv, 2(x-2)v$. These describe equations for two straight lines with positive and negative slopes respectively. As we are interested in v that simultaneously decrease both objective functions, Equation 4 is only satisfied for $v = 0$. Hence, all points for which $\mathbf{x} \in (0, 2)$ provide a solution to the auxiliary problem that is identically 0.

For $\mathbf{x} \in (-\infty, 0)$ or $\mathbf{x} \in (2, \infty)$, v is not zero. In this situation, the slopes of the straight line equations have the same sign (negative and positive respectively) and hence $||v|| = 1$ satisfies the auxiliary problem. However the step-size computation allows a large step that will move any point to either 0 or 2.

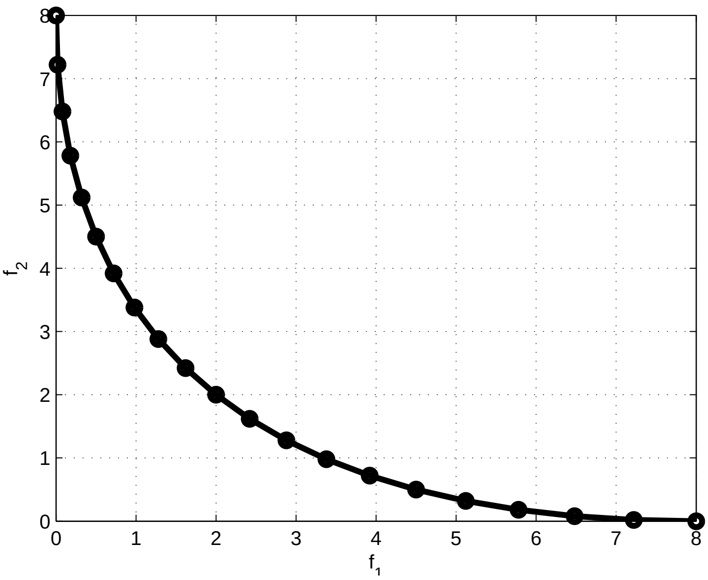
5 Convex Pareto Fronts

5.1 Uniform Pareto front

We now try to show that the algorithm can recover convex Pareto fronts when the points on it are equally-spaced [16]. We consider a simple example with two objective functions that have quadratic form in two design variables, $x_1, x_2 : f_i = (x_1 - c_i)^2 + (x_2 - c_i)^2$, $c_i = 0, 2 \;\; i = 1, 2$. Hence, f_1 is "centered" at the origin and f_2 is "centered" at $(2, 2)$. The Pareto front in objective space

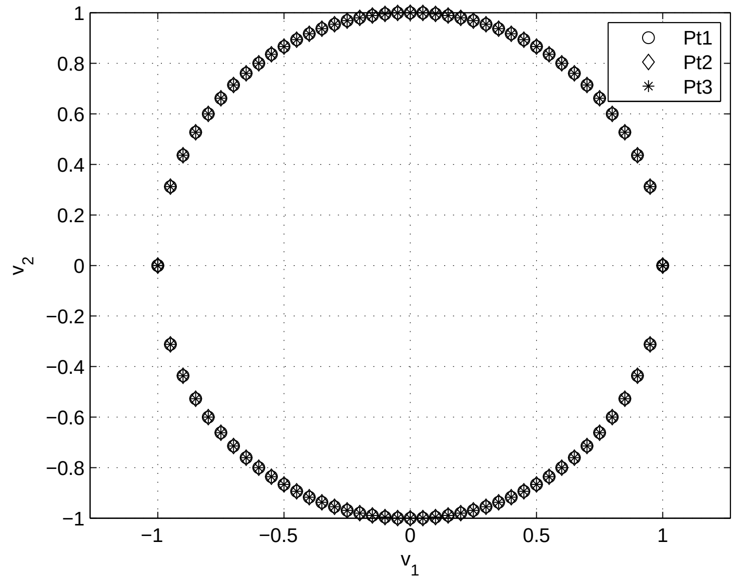


(a) Pareto front (solid line from (0,0) to (2,2)) in Design Space

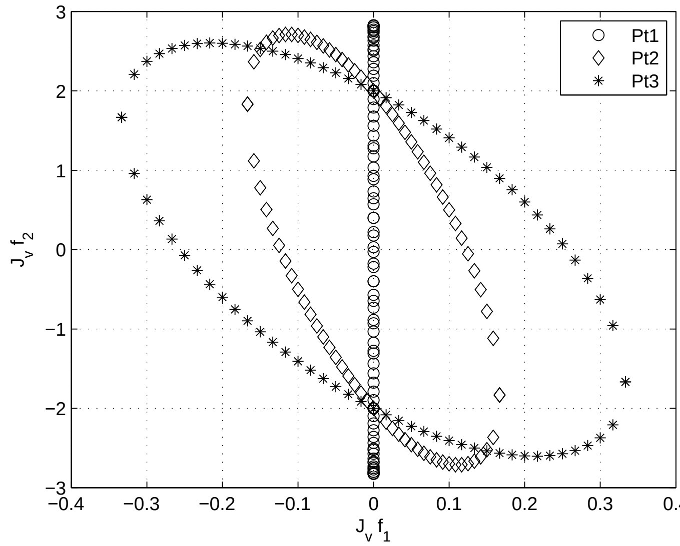


(b) Pareto front in Objective Space

Figure 2. Pareto front in Design and Objective Space



(a)



(b)

Figure 3. Components of v and $\mathscr{J}_v\mathbf{f}$ for $Pt1 = (0,0)$, $Pt2 = (0.5,0)$ and $Pt3 = (1.0,0.0)$

(f_1, f_2) is a convex curve joining $(0,0)$ to $(2,2)$ in design space. Both end points are part of the Pareto front can be obtained using single objective optimization. Along this curve $\nabla f_1 = -r\nabla f_2$. However, at the optimal solution for f_1 (or f_2), the gradient is 0 and hence the Pareto optimality condition in Equation 6 is not very useful.

The Pareto front can either be described in design space or in objective function space. For analytical problems like the one considered here, the condition in Equation 6 can be used to analytically recover the Pareto front. Using Equation 6, we can write two equations for x_1 and x_2 as

$$\begin{aligned}(x_1 - c_1) &= -r(x_1 - c_2) \\ (x_2 - c_1) &= -r(x_2 - c_2)\end{aligned} \tag{8}$$

Hence, $x_1 = x_2$ along the front which the equation for a straight-line and we know this line passes through $(0,0)$ (the minima of

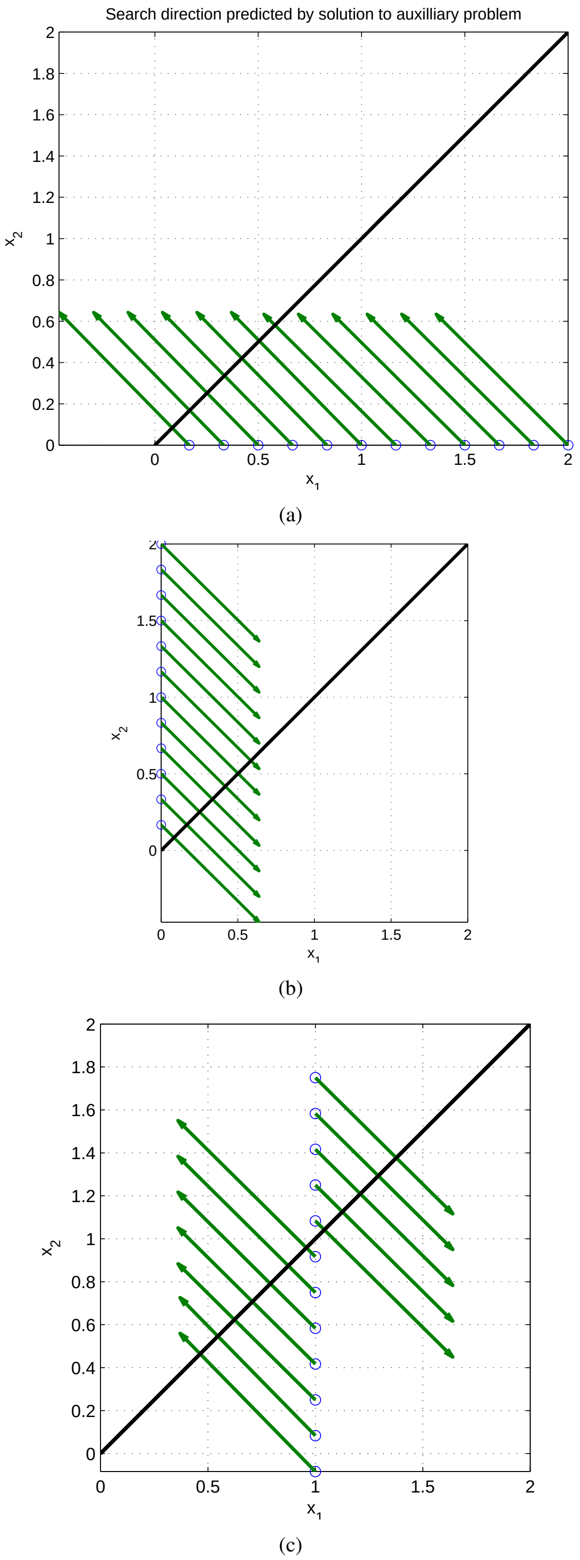


Figure 4. Solution to Equation 4 for various initial points not on the Pareto front. The Pareto front is shown as a straight line joining (0,0) to (2,2). The arrows represent the search direction predicted for the point marked with a circle.

the single-objective optimization for f_1. The resulting Pareto front in design (Figure 2(a)) and objective (Figure 2(b)) space in shown Figure 2.

The above analysis shows that if we start at the minima for one of the two objectives, moving along the direction suggested by Equation 7 enables tracking of the Pareto front. We now consider what happens when we move away from the Pareto front. Let $\mathbf{x}'$ be such a point. The gradient matrix that is used in the auxiliary problem for this point is

$$\mathscr{J}\mathbf{f} = \begin{pmatrix} x_1 - c_1 & x_2 - c_1 \\ x_1 - c_2 & x_2 - c_2 \end{pmatrix}$$

and we would like to minimize $h(v)$ under the constraint that $||v|| \leq 1$.

$$\begin{aligned} h(v) = max(& ((x_1 - c_1) * v_1 + (x_2 - c_1) * v_2), \\ & ((x_1 - c_2) * v_1 + (x_2 - c_2) * v_2)) \end{aligned} \tag{9}$$

Figure 3 show plots used in the construction and solution of the auxiliary problem for different points in the design space. Figure 3(a) shows the components of v used to determine the search direction for the auxiliary problem. Once, the gradient matrix is evaluated, the auxiliary problem is solved by sampling $h(v)$ for various values of v shown in Figure 3(a). Figure 3(b) shows the components of the vector obtained by multiplying the gradient matrix with a particular sample of v. The components of $\mathscr{J}_v\mathbf{f}$ for three different points in the design space are shown in sub-figure 3(b). We are interested in the portion where both components are negative as this leads to simultaneous reduction of both components of $\mathbf{f}$. We sample the gradient-matrix for various values of the search direction at three points in the design space labeled $Pt1, Pt2, Pt3$. $Pt1$ is on the Pareto front and the other points are along the x_1 axis while the Pareto front is a straight-line joining $(0,0)$ to $(2,2)$. These plots show that for points on the Pareto front ($Pt1$), the only solution to the auxiliary problem is $v = 0$, while at $Pt2$ and $Pt3$ there exists a search direction that points towards the Pareto front. Figure 4 shows the predicted search direction for other points in the design space. Figure 4(a) shows the search direction for points along the x-axis and to the right of the Pareto front, 4(b) shows the search direction for points along the y-axis and to the right of Pareto front and 4(c) shows the search direction when the points are distributed on either side of the front. It is clear the algorithm is capable of predicting movement towards the Pareto front for points that are not on it and also switches direction across the front.

5.2 Non-uniform Pareto front

We now consider a problem with a convex Pareto front but with non-uniform spacing [16]. Unlike the previous example, the Pareto points are not uniformly distributed. Let $f_1 = x_1$ and $f_2 = g(1-\sqrt{f_1/g})$, where $g = 1+9x_2$ pose a bi-objective problem in two design variables. As the unconstrained problem has minima at $-\infty$, we impose bounds on the range of $\mathbf{x} \in [0,1]$. The Pareto front in design space is a line joining $(0,0)$ to $(1,0)$ and the Pareto front in objective space for equi-spaced sampling along the front in design space is shown in Figure 5.

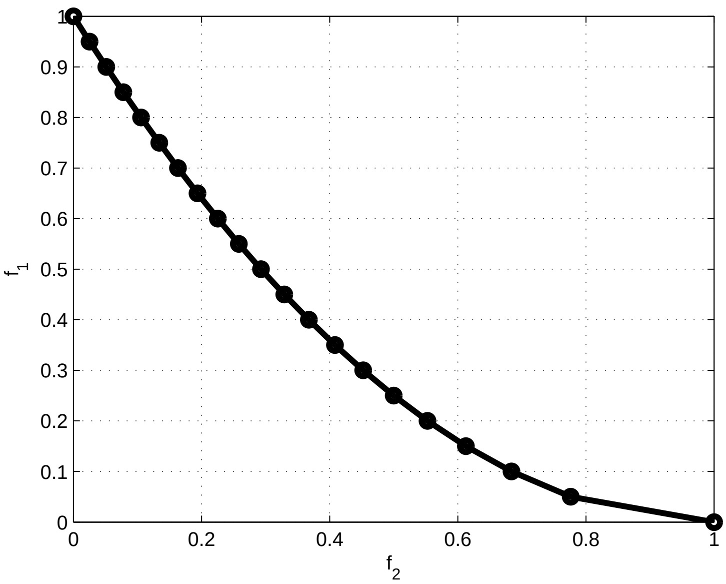


Figure 5. The Pareto front in objective space with non-uniform spacing

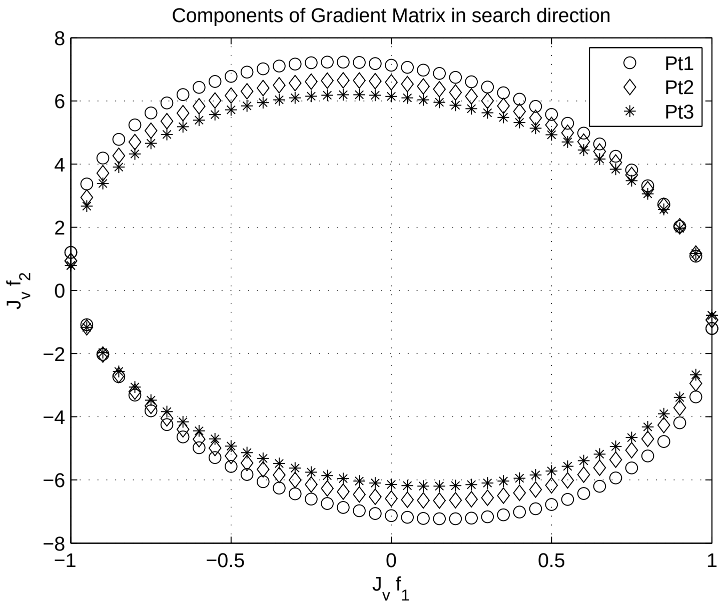
The gradient matrix can be written as

$$\mathscr{J}\mathbf{f} = \begin{pmatrix} 1 & 0 \\ -\frac{\sqrt{g/f_1}}{2} & 9(1-\sqrt{f_1/g}) + \frac{9}{2}\sqrt{f_1/g} \end{pmatrix}$$

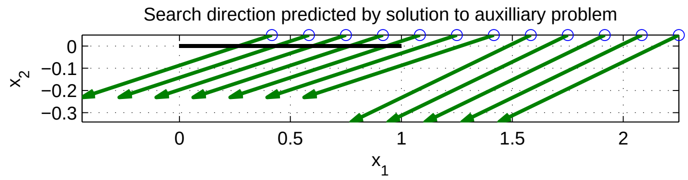
Figures 6(a) and 6(b) show plots of the components of gradient matrix and the search direction respectively at three points offset from the Pareto front. Figure 7(a) shows the non-parallel search directions near the origin reflecting possible non-uniformity in the spacing of the points on the front. Figure 7(b) shows parallel search directions near $(1,0)$, leading to more uniform distribution of Pareto critical points.

6 Concave Pareto Fronts

We now consider a problem with a concave Pareto front [16]. Let $f_1 = x_1$ and $f_2 = g(1-(f_1/g)^2)$, where $g = 1+9x_2$ pose a bi-objective problem in two design variables. As the unconstrained problem has minima at $-\infty$, we impose bounds on the range of $\mathbf{x} \in [0,2]$. The Pareto front in design space is a line joining $(0,0)$

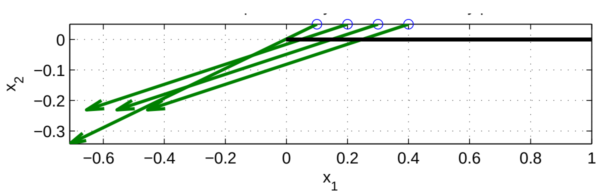


(a)

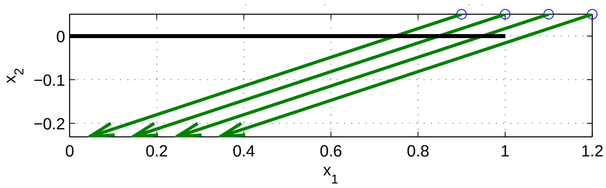


(b)

Figure 6. Components of the gradient matrix for convex non-uniform Pareto front for for $Pt1 = (0.25, 0.05)$, $Pt2 = (0.4, 0.05)$ and $Pt3 = (0.6, 0.05)$. The Pareto front is shown as a straight line joining (0,0) to (1,0). The arrows represent the search direction predicted for the point marked with a circle.



(a) Change in search direction to reflect non-uniform spacing



(b) Parallel search directions to reflect uniform spacing

Figure 7. Change in search direction at either ends of the Pareto front.

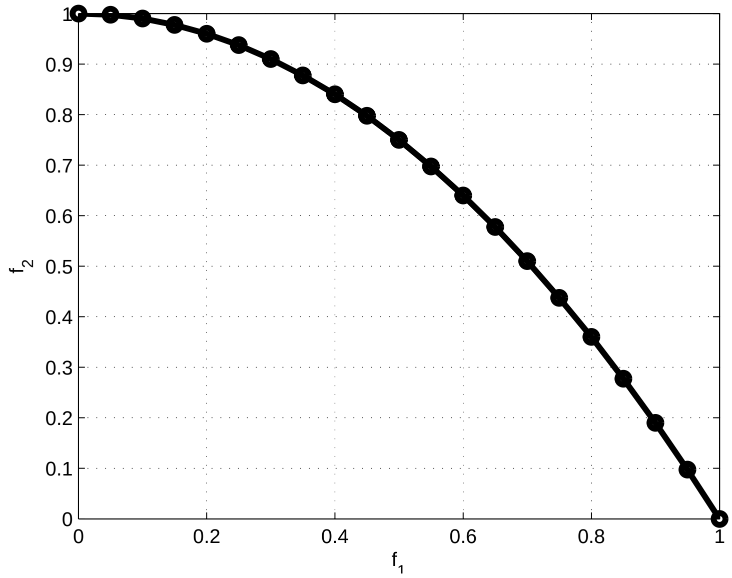


Figure 8. The concave Pareto front in objective space

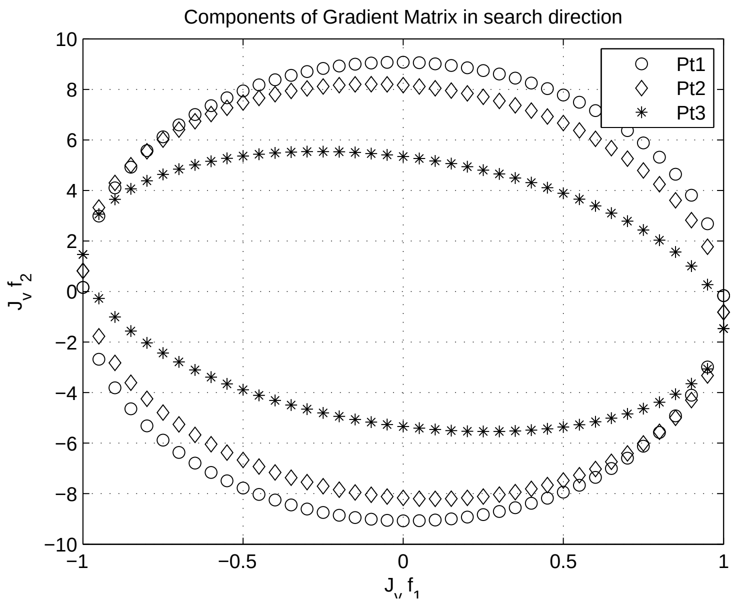


Figure 9. Components of v and $\mathscr{J}_v\mathbf{f}$ for $Pt1 = (0.1, 0.025)$, $Pt2 = (0.5, 0.025)$ and $Pt3 = (0.9, 0.025)$. All three points lie above the Pareto front and hence should have a non-zero search direction.

to $(1,0)$ and the Pareto front in objective space for equi-spaced sampling along the front in design space is shown in Figure 8. Figure 9 shows the search direction and the components of the gradient vector for the auxiliary problem at three points offset from the Pareto front. Figure 10 shows the search direction in design space for a uniform sampling of points. The gradient vectors are roughly parallel suggesting a uniform Pareto front.

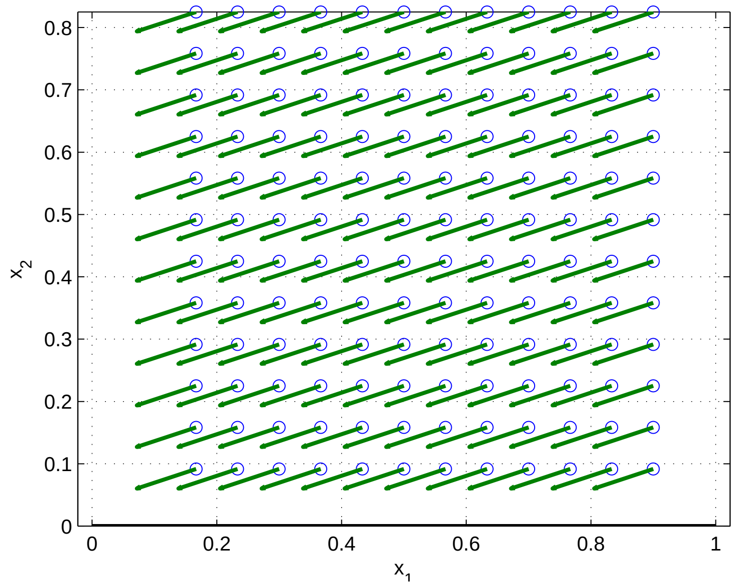


Figure 10. Search direction from the auxiliary problem for a uniform sampling of the relevant design space

7 Results for a Turbo-machinery Problem

We now apply the algorithm that is the subject of this paper to a turbo-machinery problem. We study the trade-off between aerodynamics and mechanical integrity for a turbine blade. Due to proprietary reasons we cannot show detailed plots of the geometry or mention the operating conditions of this blade. We will try to elucidate the salient features of the blade as pertinent to the algorithm described in this study. In particular, we are interested in maximizing the mid-span aerodynamic efficiency while ensuring that the mechanical frequency of interest is above a particular value. The baseline blade had an acceptable aerodynamic efficiency but one of the torsional frequencies was close to resonance. As there exists a trade-off between the aerodynamic and structural performance, the design exercise was to determine an alternate point on the trade-off front which had acceptable aerodynamic and structural performance.We use a two-dimensional viscous solver, MISES, for the aerodynamic analysis and ANSYS for the modal (structural) analysis. The structural model used for the design study uses a variety of sophisticated boundary conditions to describe the operation of the blade and the geometry includes the blade, cooling holes and detailed description of the dovetail connection to the shaft. When appropriate, the numbers for aerodynamic efficiency are non-dimensionalized with the baseline value but the numbers for the mechanical frequency will always be shown as a percentage change from the baseline value.

We study this problem with two approaches to understand their merits and demerits. The first one is a DOE-based approach which involves a) sampling of the design space with a particular

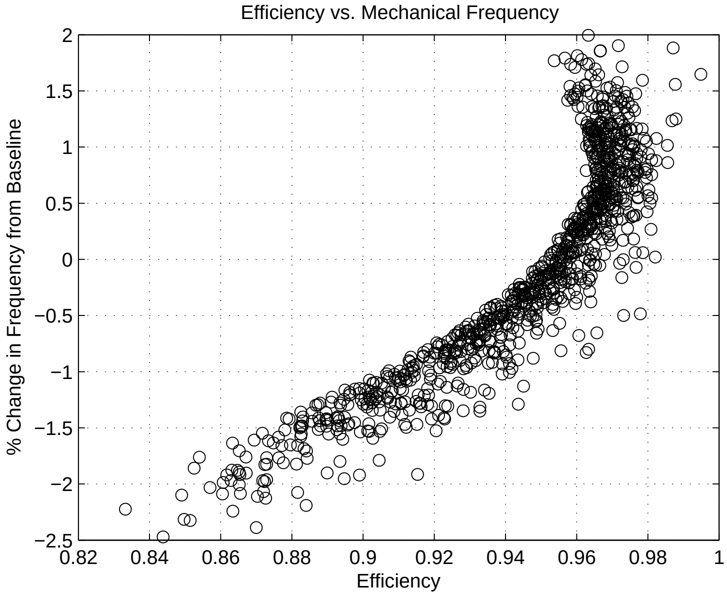


Figure 11. Monte Carlo Points predicted with surrogate model

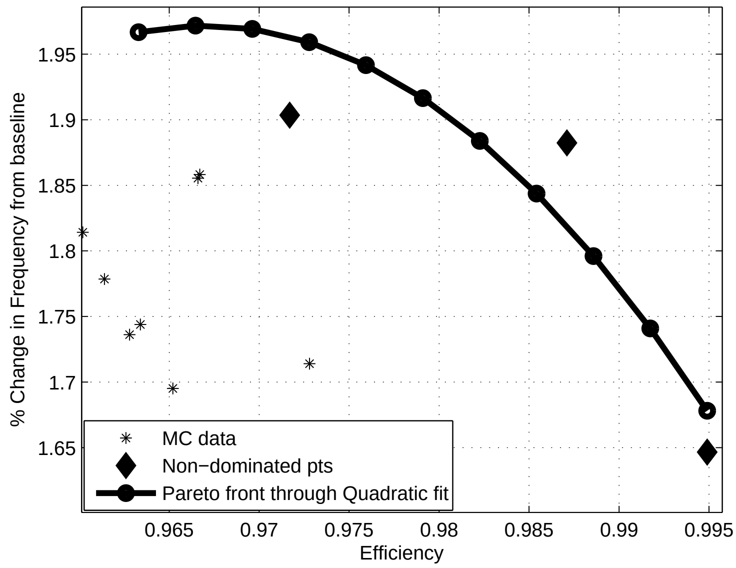


Figure 12. Zoomed-in view of the non-dominated points and Quadratic fit for approximate Pareto front.

DOE, b) construction of a regression equation from the sampling points, c) evaluation of this regression equation through a Monte-Carlo simulation to obtain a cloud of points in objective space, d) identification of non-dominated solutions from this cloud of points and labeled to be on the Pareto front and e) to complete the design study, verification of these points using MISES and ANSYS to validate the predicted performance from the regression equation. In the second approach, a gradient-based one, we start with the baseline, and first solve a SOOP to determine the starting point for the Pareto tracking algorithm. From this starting point, the Pareto-tracking method described in Algorithm 1 is applied to recover the other Pareto points. The collection of these points then describes the Pareto front.

Five design variables, namely stagger, cross-sectional area, trailing edge feature, position of maximum thickness and radius of the leading edge are used as design variables for this study. For ease of exposition, we discuss the two dominant design variables, namely the stagger angle of the blade and the position of maximum thickness. The DOE-based approach and the gradient-based approach were executed in non-dimensional design space. A ± 1 non-dimensional change in stagger angle corresponds to $\pm$ 4 degree change around the baseline and a ± 1 non-dimensional change in maximum thickness corresponds to a change by $\pm$ 10% of the blade chord at mid-span. Positive changes in stagger correspond to increasing stagger and positive changes in position of maximum thickness correspond to movement of the location of the maximum towards the trailing edge.

7.1 DOE-based Approach

With the 5 design variables, an Orthogonal Latin-Hypercube (OLH) sampling technique was used to evaluate 50 points in the design space to construct the surrogate model. The choice of 50 is based on a rough rule-of-thumb where 10 points are used for each design variable to provide satisfactory coverage of the design space along with good quality regression surfaces. An alternate, less expensive DOE could have been chosen but MISES-based simulations are known to result in noisy estimates and hence require more simulations than that required to estimate the unknown coefficients in a surrogate model. From these 50 simulations, a quadratic regression surface was constructed. This regression equation was then polled using a Monte-Carlo (MC) simulation process to obtain 10,000 evaluations. Please note that we do not use an evolutionary algorithm that can tailor successive populations to track the Pareto front.

Figures 11 and 12 show the Pareto front between aerodynamic efficiency and structural performance obtained from steps a)-d) of the DOE-based approach. For clarity, only a tenth of the MC points are shown. These points are then sorted for non-dominance using a simple $O(n^2)$ sorting algorithm. Figure 12 shows the selected non-dominated points from a population produced from the meta-model. As there were not enough points along the Pareto front, an approximate Pareto front was estimated from the non-dominated points. To obtain this approximate front in objective space, a quadratic polynomial was fit through the non-dominated points. 10 points on this approximate fit were selected. Their corresponding points in design space was approximated using a linear interpolation routine. Both the sparsity of the non-dominated points and its irregularity is evident from this plot. This suggests that 1) instead of using the MC approach to determine the front, one should use an evolutionary algorithm that tailors its population to track the Pareto front and 2) to get good results with the MC approach, the quality of the meta-model fit needs to be improved. Upon further inspection, it was observed that the R^2 value of the fit for the change in frequency was around 0.9959 and the poor fit was for the aerodynamic efficiency. It is not uncommon for MISES simulations to exhibit lack of convergence leading to low R^2 values for the fits. In spite of these shortcomings, we used this data as computed. The Pareto front is (under the validity of the quadratic fit) concave and potentially not equally spaced. On inspection of the points from the OLH sampling and along the Pareto front the following observations about the design space were made:

1. Among the 5 design variables, stagger and position of maximum thickness were the dominant design variables. Between stagger and position of maximum thickness, stagger had a greater impact on aerodynamic and structural performance. While stagger did not have a large interaction effect with the other design variables, the position of maximum thickness was weakly correlated to cross-sectional area and radius of leading edge.
2. While the baseline geometry had good aerodynamic performance, it was possible to increase the aerodynamic efficiency by reducing the stagger of the blade. This suggests that the incidence to the baseline blade was less than optimal from an aerodynamic perspective. From a structural perspective, larger improvements were obtained by reducing the stagger even further than that suggested by the aerodynamic performance measure. This phenomenon is a little harder to explain due to the complicated geometry used for the structural analysis. It seems that this reduction in stagger along with smaller changes to the other design variables leads to improved stiffness against the frequency of concern. Additionally, reducing the stagger below that suggested by the aerodynamic measure to produce improvements in structural performance clearly points to the possibility of a Pareto-like trade-off.
3. The position of maximum thickness has a bigger effect for aerodynamic performance than for the structural performance. The combination of reducing the stagger and marginal positive movement (towards the trailing edge) of the position of the maximum thickness results in aft-loaded airfoils whose predicted efficiency is higher. However, it must be pointed out that these aft-loaded airfoils observed in the DOE included non-zero changes to the other three design variables namely, cross-sectional area, trailing edge feature and radius of the leading edge.

7.2 Gradient-based Approach

The gradient-based algorithm was started from the baseline geometry which had an efficiency of 0.9503. All 5 design variables listed previously were used in a SOOP optimization driven by forward difference gradients and a steepest-descent algorithm with a constant step-size with the objective of maximizing efficiency. It took two iterations of the optimizer to reach the optimum and the cost was 12 simulations of MISES and ANSYS (6 simulations for each iteration). The successive evolutions of the SOOP is shown in Figure 13 where iteration 0 is the baseline. Further improvements in efficiency after 2 iterations was minimal (less than 0.1 pts) and hence the optimization process was stopped after 2 iterations. After these two iterations, the efficiency was improved by ≈ 1.3 pts, increasing from 0.9503 to 0.9640. The optimized geometry had a smaller stagger angle than the baseline (≈ 2.55 degrees smaller) and negative changes in the position of maximum thickness ($\approx 1\%$ of blade chord). The reduction in stagger is similar to the trend observed in the DOE-based approach. The baseline geometry had a less than optimal incidence to the incoming flow and the reduction in stagger aligns the leading edge of the blade to the incoming flow. This was further confirmed in the loading profiles which had a smaller leading edge suction peak. Along with this change in the stagger angle, the position of maximum thickness moves slightly towards the leading edge and the combination (along with small changes to the other 3 design variables) results in a more aft-loaded blade. These changes also seem to help the structural performance as the frequency of interest has increased by $\approx 1.5\%$ from the baseline. Also, it is interesting to note that the non-dominated point with the highest efficiency predicted with the surrogate model is close to 0.995. This would suggest that the solution to the SOOP problem did not predict an optimal design. However, as seen in Figure 14, when this point was evaluated with MISES, the prediction is closer to that predicted by the gradient-based algorithm suggesting that the surrogate for aerodynamic efficiency is poor as already indicated by the R^2 value of the fit. The 10 points obtained through a quadratic fit of the non-dominated points were used in verification runs and the Pareto front thus determined is also shown in Figure 14.

From the optimum predicted by the solution to the SOOP, the Pareto tracking algorithm in Section 4 was started. This starting Pareto point was perturbed by an (arbitrary) amount of 0.05 (in non-dimensional units) for each design variable. From this perturbed point, the auxiliary problem was solved, using finite-difference gradients, to recover the next Pareto point. This process of perturbing the new Pareto point and solution of the auxiliary problem to *snap* to the Pareto front was repeated for 9 points. Typically, only one gradient evaluation was required for the auxiliary problem before a new point on the Pareto front was obtained. So in essence, the cost of this method is equal to the cost of the first single objective optimization problem plus the cost of the gradient evaluation for each of the required Pareto

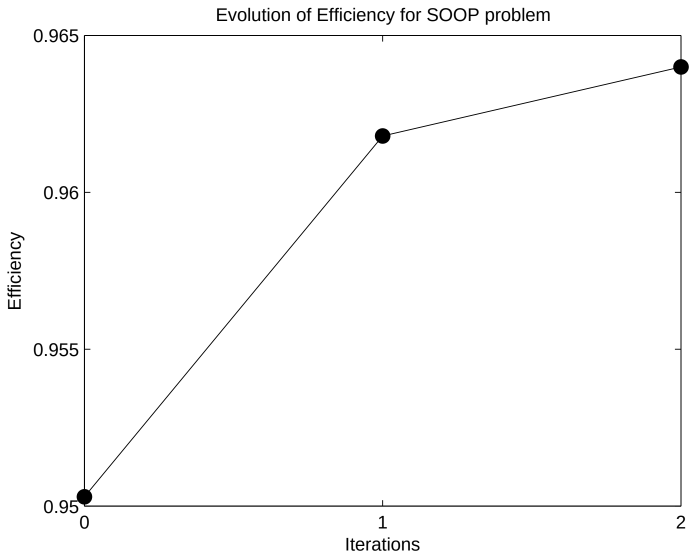


Figure 13. Evolution of the aerodynamic efficiency during solution of the SOOP.

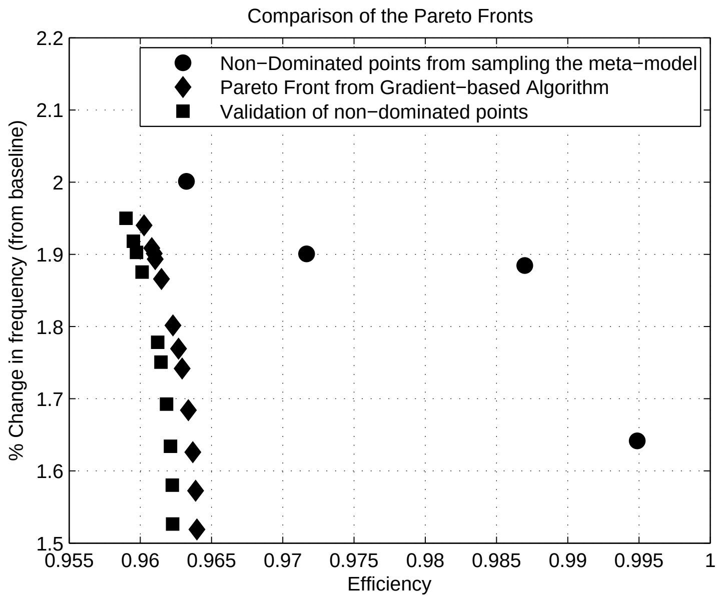


Figure 14. Comparison of the Pareto fronts. Note that "front" obtained from the meta-model is different from that obtained when the high-fidelity model is used to verify the front. The verified front lies closer to that obtained with the gradient-based algorithm.

front points. Here, all gradients were determined using forward-difference approximations. The overall cost of the entire algorithm was equivalent to 66 (12 for the SOOP plus 54 for the 9 remaining points on the Pareto front) simulations of MISES and ANSYS compared to roughly 60 with the DOE-based approach. If we had only required 5 points on the Pareto front, the cost for the DOE-based approach would have been 55 evaluations as opposed to 36 for the gradient-based approach. Furthermore, the major cost of the gradient-based approach is the cost of the gradient evaluations. It is possible to make this cost independent of the number of design variables using the adjoint method thereby providing more savings through the gradient-based approach.

Figure 14 shows the 10 points recovered by the gradient-based Pareto tracking algorithm. The Pareto front suggests that it is possible to trade ≈ 0.35 pts in aerodynamic efficiency for about a 0.5% increase in the mechanical frequency of interest. As we move along this Pareto front starting from the optimum of the SOOP, the stagger angle continuously reduced. Relative to the baseline, the stagger angle was reduced by ≈ 2.55 degrees for the optimal point of the SOOP and by ≈ 4.2 degrees at the other end of the Pareto front. The former corresponds to a non-dimensional change of approximately -0.64 and the latter approximately -1. The continuous decrease of stagger angle along the Pareto front results in increasing off-incidence loses that degrade the aerodynamic efficiency. It however seems to have a positive impact on the structural performance. Along the front, the other design variables move in a manner that makes the airfoils more aft-loaded thereby offsetting some of the aerodynamic performance penalty incurred by the change in stagger angle. For example, the non-dimensional change in position of maximum thickness varied from -0.089 to 0.25 along the Pareto front. Figure 14 shows that the Pareto front obtained by the gradient-based algorithm. For comparison, the original non-dominated points from MC simulations with the surrogate model and the verification of the approximate Pareto front are also shown. The Pareto front as predicted by the gradient-based algorithm is very close to those obtained from the verification runs. The movement of the Pareto front during verification runs was one of the primary reasons we started investigating alternate algorithms to directly track the Pareto front with high-fidelity simulation tools.

Finally, to additionally verify the gradient-based algorithm, the solution to the auxiliary problem was computed as different points in design space of stagger and position of maximum thickness. This requires gradient evaluations at a set of selected points. Here we have sampled the two dimensional design space at 256 points at 16 equal intervals over the range for non-dimensional change in stagger and position of maximum thickness. Figure 15 shows the solution to the auxiliary problem at these sampling points. The (approximate) Pareto front is shown with a solid line. Along this line the auxiliary vectors point in parallel but opposite directions is also shown. This shows that the Pareto front is embedded in a small portion of the design space. More importantly the extent of the line is in very close agreement with the changes in stagger and maximum thickness along the Pareto front recovered in Figure 14. The region to right of $x_1 \approx -0.6$ shows the "gradient" vector pointing away from

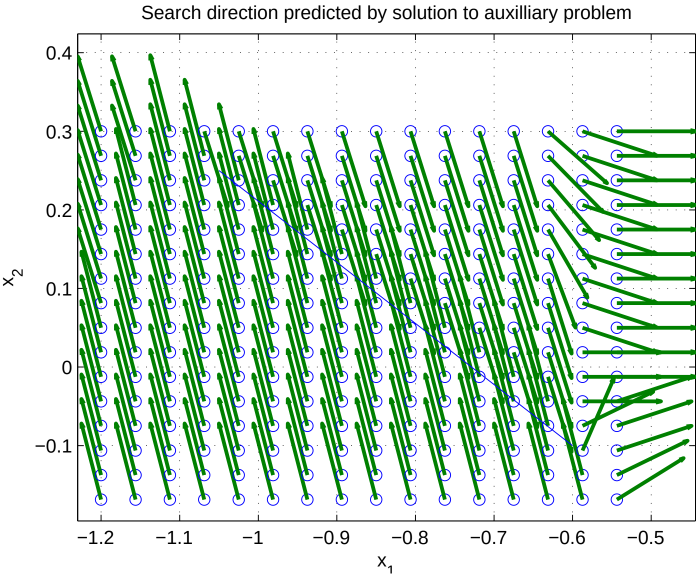


Figure 15. Search direction from the auxiliary problem for a uniform sampling of the relevant design space. The solid line shows the approximate position of the Pareto front in design space. The x-axis is the non-dimensionalized change in stagger angle and the y-axis is the non-dimensionalized change in position of the maximum thickness.

the Pareto front. Hence, around this region, the gradient-based approach will be sensitive to the choice of "steps" taken while marching along the Pareto front. This is useful information as it points to potential issues towards making this algorithm robust in practical Engineering applications.

Method	Cost	Function Evaluations
DOE - OLH	Meta-model	$\approx 10\,X \times (\sum_{i=1}^{Y} a_i)$
	Sampling	≈ 0
	Pareto Evaluation	$= P \times (\sum_{i=1}^{Y} a_i)$
DOE - FCC	Meta-model	$= 2^{X+1} \times (\sum_{i=1}^{Y} a_i)$
	Sampling	≈ 0
	Pareto Evaluation	$= P \times (\sum_{i=1}^{Y} a_i)$
Grad - FD	SOOP	$= q \times (X+1) \times (\sum_{i=1}^{Y} a_i)$
	Pareto Tracking	$= P \times (X+1) \times (\sum_{i=1}^{Y} a_i)$
Grad - Adjoint	SOOP	$= q \times 1.5 \times (\sum_{i=1}^{Y} a_i)$
	Pareto Tracking	$= P \times 1.5 \times (\sum_{i=1}^{Y} a_i)$

Table 1. Computational cost of different approaches. OLH is Orthogonal Latin-Hypercube Sampling, FCC is Face-Centered Composite Sampling and FD is Forward Difference. The number of design variables is represented by X, the number of objectives is represented by Y and the desired number of points that we wish on the Pareto front is given by P. The cost for the function evaluation of each objective is given by a_i, $i = 1, \ldots Y$ and q is the number of iterative steps required for the solution of the single objective problem. To evaluate the cost using the Adjoint approach, it is assumed that the cost of the adjoint calculation is around 1.5 the cost of the function evaluation and there exists the adjoint capability for all objectives. Typically, the aerodynamic gradients are the most expensive to evaluate and it suffices that an adjoint module exists for this sensitivity evaluation.

8 Conclusions

We have utilized a gradient-based algorithm for unconstrained multi-objective problems to recover Pareto points and the Pareto front. The method has been shown to work well for convex (with non-uniform spacing) and concave fronts. Though the results have not been reported here, it has also been used for an academic problem with a disjoint Pareto front but it requires enhancements to the current algorithm that enables the algorithm to execute "discontinuous jumps" at the edges of Pareto fronts. Finally, we have used it for a multi-disciplinary problem in axial turbines. These studies have tested the feasibility of the approach and show that they can overcome some of the difficulties inherent in scalarization (or weighted sum) approaches. The applicability of the method seems to be problem (and capability) dependent. Table 1 captures the relative cost of different approaches to MOOP. For problems with a small number of design variables, a large number of objectives and where the determination of a good meta-model is easy, an evolutionary algorithm seems to more efficient. Even if a good meta-model can be obtained, the method outlined here can be used along with the analytical gradients from the meta-model to determine the portion of the design space where the solution to the auxiliary problem results in a vector of magnitude 0. These points will be the Pareto points. The auxiliary problem has also been used in mutation operators of evolutionary algorithms by [17] to determine the optimal way to distribute and evolve population along the Pareto front. For problems where the required number of Pareto points is small and the determination of a good meta-model is difficult, the gradient-based approach presented here seems to be a better candidate. Finally, if adjoint modules are available for the most computationally expensive portions of the simulations and the problem has a large number of design variables and relatively few objective functions, the gradient-based approach is clearly the better choice of methods. In the future, we hope to include practical considerations to the present algorithm, namely constraints and acceleration techniques to reduce the overall turn-around time of Algorithm 1.

9 Acknowledgment

The authors would like to thank General Electric Company for permission to publish this document. We would also like to thank 3 anonymous referees whose input greatly helped improve the presentation of this paper.

REFERENCES

[1] Deb, K., Multi-Objective Optimization using Evolutionary Algorithms, John Wiley and Sons LTD., New York, 2001.

[2] Zitzler, E., Deb, K., and Thiele, L., Comparison of Multi-Objective evolutionary algorithms: Empirical results, Evol. Comput.,8, 125-148, 2000.

[3] Marler, R. T. and Arora, J. S., Survey of Multi-Objective Optimization Methods in Engineering, Structural and Multidisciplinary Optimization, 26, 6, 269-295, 2004.

[4] Stadler, W., A Survey of Multicriteria Optimization, or the Vector Maximum Problem, Journal of Optimization Theory and Applications, Vol. 29, pp. 1-52, 1979.

[5] Steuer, R., Multiple Criteria Optimization - Theory, Computation and Application, Krieger Publishing Company, January 1986.

[6] de Weck, O. L. and Kim, I. Y., Multi-Objective Optimization via the Adaptive Weighted Sum Method, 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Albany, New York, August30-September 1, 2004.

[7] Kim, I. Y. and de Weck, O. L., Adaptive weighted-sum method for bi-objective optimization: Pareto front generation, Structural and Multidisciplinary Optimization, September, 2004.

[8] Das, I. and Dennis, J., Normal-Boundary Intersection: A New Method for Generating Pareto Optimal Points in Multicriteria Optimization Problems, SIAM Journal on Optimization, Vol. 8, No.3, pp. 631-657, 1998.

[9] Jin, Y., Olhofer, M. and Sendoff, B., Dynamic Weighted Aggregation for Evolutionary Multi-Objective Optimization, Proceedings of the Genetic and Evolutionary Computation Conference, GECCO, July, San Franscisco, CA, 2001.

[10] Fliege, J. and Svaiter, F. B., Steepest Descent Methods for Multicriteria Optimization, Mathematical Methods of Operations Research, Vol. 51, No. 3, pp. 479-494, 2000.

[11] Sevastyanov, V., Gradient-based Algorithms for Multi-Objective Optimization Problems, 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Dallas, TX, 2010.

[12] Levitan, L. and Sevastyanov, V., Exclusion of Regions Method for Multi-Objective Optimization, U.S. Patent, Filed, April 28, 2005.

[13] Drela, M. and Youngren, H., A User's Guide to MISES 2.63, MIT Aerospace Computational Design Laboratory, February 2008

[14] ANSYS, Release 10.0.

[15] Pereyra, V., Saunders, M. A. and Castillo, J., Equispaced Pareto front construction for constrained biobjective optimization, Systems Optimization Lab, Report SOL 2010-1, Stanford University, Stanford, CA, 2010.

[16] Parsopoulos, K. E. and Vrahatis, M. N., Particle swarm optimization method in Multi-Objective problems, Symposium on Applied Computing, Proceedings of the 2002 ACM symposium on Applied computing, pp. 603-607, Madrid, Spain, 2002.

[17] Bosman, A. N. and Jong, E. D., Exploiting Gradient Information in Numerical Multi-Objective Evolutionary Optimization, Proceedings of the Genetic and Evolutionary Computation Conference, GECCO, June, Washington DC, 2005.