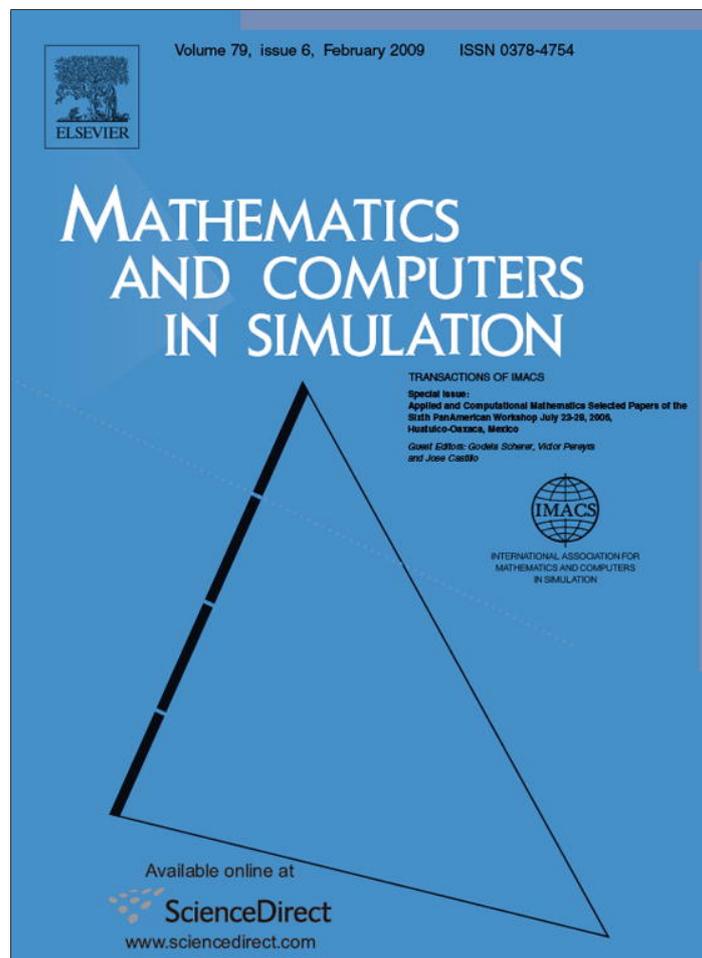


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Fast computation of equispaced Pareto manifolds and Pareto fronts for multiobjective optimization problems[☆]

Victor Pereyra^{*}

Weidlinger Associates Inc., 399 W. El Camino Real #200, Mountain View, CA 94040, United States

Received 29 August 2006; received in revised form 27 December 2006; accepted 20 February 2007

Available online 25 February 2007

Abstract

In this paper, we consider the problem of generating a well sampled discrete representation of the Pareto manifold or the Pareto front corresponding to the equilibrium points of a multi-objective optimization problem. We show how the introduction of simple additional constraints into a continuation procedure produces equispaced points in either of those two sets. Moreover, we describe in detail a novel algorithm for global continuation that requires two orders of magnitude less function evaluations than evolutionary algorithms commonly used to solve this problem. The performance of the methods is demonstrated on problems from the current literature. © 2007 IMACS. Published by Elsevier B.V. All rights reserved.

Keywords: Pareto manifolds; Pareto fronts; Multiobjective optimization

1. Introduction

Multi-objective optimization is becoming a common tool in Engineering and Scientific applications. Most optimization problems in industry are multi-objective, non-linear, constrained and multi-modal, i.e. very tough.

Multi-objective optimization has a long history, especially in areas such as decision and game theory. It is only in the last 20 years that it has become more popular in Engineering and Scientific applications. Many books describe a myriad of approaches to the “solution” of these problems [4,23,10,8,22,12]. The quotes are used to emphasize that the most common situations lead to a continuum of equilibrium points, which in the absence of additional information are all valid “solutions” to the problem. That is where many of the different approaches appear, attempting to decide which of these infinity many possibilities should be chosen. Thus, many methods use additional “expert” or subjective information to guide the search towards a particular solution [13,21].

In recent times, with the increase in computing power, it has become possible to attempt to generate a complete discrete approximation to the Pareto manifold (set of equilibrium points) or its image by the objectives, the Pareto front. Some of these methods use escalarization as their basic tool, i.e. minimization of a linear combination of the objectives is employed to generate Pareto points. However, since the mapping from weight space to input or output space can be very nonlinear, different strategies need to be devised to obtain a well sampled set of solutions.

[☆] Research supported by the National Science Foundation under SBIR Grant # DMI-0321420, Phase IIB.

^{*} Corresponding author. Tel.: +1 650 230 0210.

E-mail address: vpereyra@yahoo.com.

Evolutionary and genetic algorithms (GA) are one class of powerful methods that has been favoured in recent times for their robustness, specially in the versions that permit to calculate a discrete representation of the Pareto manifold [3]. The negative side to these methods is the number of function evaluations required to obtain a reasonable accuracy, which grows exponentially with the dimension of the design parameter space. This is totally inadequate for realistic high fidelity design applications, where function evaluations can be very costly. They still can be useful if one replaces these expensive evaluations by surrogates. These methods do not use linear aggregation.

In this paper, we show that the Pareto manifold for a convex bi-objective problem can be approximated by solving numerically a two-point boundary value problem and from this insight we mimic techniques for the solution of such problems to obtain a continuation method that updates a whole discrete representation of the Pareto manifold while maintaining and even spacing between solutions. As an additional bonus this procedure is easily parallelizable. We also consider for comparison a simple continuation procedure and an enhanced version that includes a equispacing constraint. We will see that this last method is capable of obtaining equispaced points in the non-convex parts of the Pareto front and it is also useful to initialize the global method when this fails to converge from poor initial guesses, specially for highly curved Pareto fronts. The simple implementations we have devised to produce the results of this paper are not production codes but give us a quick way to show that this approach works in some test cases taken from the literature.

In [2], a method that is not based on GA is presented. This method is most akin to ours in terms of economy of function evaluations. It is pointed out in that paper that it is very difficult to guarantee an uniform spread of points in the Pareto manifold or in the Pareto front when using the popular method of solving many single optimization problem for convex combinations of the objectives. We will show that the methods of this paper do just that without any guessing and in a manner as economical as the method proposed in that reference, but with a simpler algorithm.

The full front continuation described in this paper does not suffer from the disadvantages pointed out in [2], with regards to usual initial value continuation, such as that of [18].

We discuss in this paper the unconstrained case to understand the geometry of the problem and that of the optimality conditions. In a companion paper, we will extend these ideas to the constrained case. The methods extend naturally to problems with more than two objectives.

2. Optimality conditions

The multiobjective optimization problem is defined as:

$$\min_{\mathbf{x} \in D} \mathbf{F}(\mathbf{x}), \tag{1}$$

where $\mathbf{x} \in R^n$, $\mathbf{F} \equiv \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\} \in R^k$ and $D \subset R^n$ is defined by a set of constraints.

One of the main features that distinguishes multi-objective from single-objective optimization is that it is very unlikely that there will be an \mathbf{x}^* that is a minimizer for each and every objective. In fact, these problems are characterized for the requirement of a subjective trade-off between conflicting objectives. The optimality concept here is known as *Pareto (or Nash, Edgeworth) equilibrium*, which in words establishes that \mathbf{x}^* is a global Pareto equilibrium point if there is no other point that is *dominated* by \mathbf{x}^* .

A point \mathbf{x} is dominated by a point \mathbf{y} iff $f_i(\mathbf{y}) \leq f_i(\mathbf{x})$, with strict inequality for at least one of the objectives. Thus, a global Pareto equilibrium point is such that no improvement for all objectives can be achieved by moving to any other feasible point. A local version of this concept is obtained if we limit the movement to an open neighborhood around the optimal point.

For differentiable convex objectives we can use the usual geometrical concepts of single objective optimization to arrive to an analytical characterization of the local Pareto points. This is easily seen first for the unconstrained case of two objectives and two independent variables. Let \mathbf{x}_1^* , \mathbf{x}_2^* be local minima of f_1 and f_2 , respectively. Then a segment of the *Pareto manifold* is a curve that joins these two points and is defined as the parametric set of solutions $x^*(\lambda)$ of the first order necessary optimality condition:

$$\mathbf{G}(\mathbf{x}(\lambda); \lambda) = (1 - \lambda)\nabla f_1(\mathbf{x}) + \lambda\nabla f_2(\mathbf{x}) = 0, \quad 0 \leq \lambda \leq 1. \tag{2}$$

Geometrically, (2) says that a point is Pareto optimal if the contours of the two objectives are tangent at it, with gradients pointing in opposite directions; i.e. the two functionals have no descent directions in common (see Fig. 1 for a cartoon description). The image by \mathbf{F} of the Pareto manifold is called the *Pareto front*.

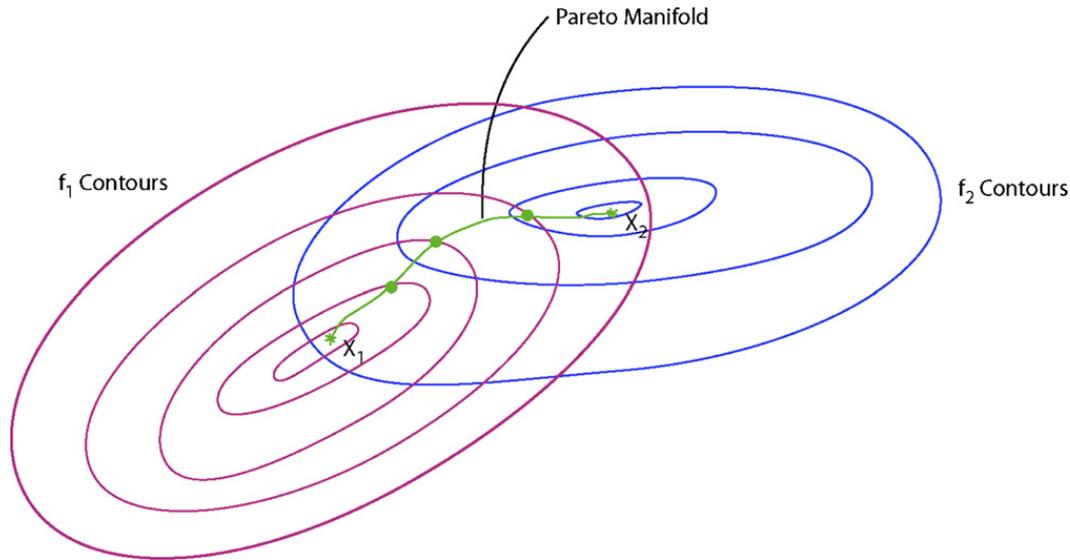


Fig. 1. Geometrical description of the Pareto manifold for a bi-objective problem with two independent variables.

3. A Differential equations approach to finding a discrete representation of the Pareto manifold

By differentiating \mathbf{G} with respect to λ we obtain:

$$\mathbf{x}_\lambda = -\mathbf{H}^{-1}(\mathbf{x})\mathbf{G}_\lambda(\mathbf{x}), \quad \text{subject to } \mathbf{x}(0) = \mathbf{x}_1^*, \quad \mathbf{x}(1) = \mathbf{x}_2^*. \quad (3)$$

This is a two-point boundary value problem for a set of n first order ordinary differential equations in the independent variable λ . Here $\mathbf{G}_\lambda = \nabla f_2(\mathbf{x}) - \nabla f_1(\mathbf{x})$, $\mathbf{H}(\mathbf{x}) = \mathbf{G}_\mathbf{x} = (1 - \lambda)\mathbf{H}_1(\mathbf{x}) + \lambda \mathbf{H}_2(\mathbf{x})$, with \mathbf{H}_i the Hessian matrix of f_i . In order for this global result to hold we must have of course that $\mathbf{H}(\mathbf{x}(\lambda))$ is nonsingular along the Pareto manifold. For the non-convex case, segments of the Pareto manifold will join each pair of minima of the functionals f_i .

In order to obtain a discrete representation of a segment of the Pareto manifold we can solve problem (3) numerically. Since we also want an uniform sampling of this manifold segment, we can require a mesh refinement equidistribution of the arc length as additional constraints. An initial approximation to the trajectory can be obtained by the straight line joining the two minima:

$$\mathbf{x}_i^{(0)} = \mathbf{x}^{(0)}(\lambda_i) = (1 - \lambda_i)\mathbf{x}_1^* + \lambda_i\mathbf{x}_2^*, \quad \lambda_i = i * \delta\lambda, \quad i = 0, \dots, l + 1. \quad (4)$$

Unfortunately, we will need still another derivative to apply Newton's method to solve the resulting nonlinear system of algebraic equations that arises when using global discretization methods to solve this 2PBVP.

Although this is an elegant approach that sheds some light on the potential structure of the Pareto manifold, it may not be too practical, since it requires third order derivatives, so we go back to (2) and consider standard continuation [19], combined with a direct application of Newton's method, in case we have the required differentiability.

Starting from \mathbf{x}_1^* , i.e. $\lambda = 0$, we solve successively $\mathbf{G}(\mathbf{x}(\lambda_i); < MML : MSPACECLASS = "N BSP" / > \lambda_i) = 0$, with λ_i defined as above, although an adaptive algorithm is also possible and probably wise (see for instance, [9]). Newton's method applied to the above equation for a given λ_i can be written as:

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} - \mathbf{G}_\mathbf{x}^{-1}(\mathbf{x}^{(j)}(\lambda_i)) \mathbf{G}(\mathbf{x}^{(j)}(\lambda_i)), \quad j = 0, \dots$$

Of course, we write the above iterative step as a system of linear equations with matrix $\mathbf{G}_\mathbf{x}$. Singularity of $\mathbf{G}_\mathbf{x}$ indicates possible bifurcation of the solution path. Another potential problem, turning points, should not affect this parametrization (arc length). If $\mathbf{x}^*(\lambda_i)$ is the approximate solution of the above equation, then we use it as initial value for the next one: $\mathbf{x}^{(0)}(\lambda_{i+1}) = \mathbf{x}^*(\lambda_i)$.

For bi-objective problems one can also use a general continuation code for solving systems of nonlinear equations, such as [19], or one that applies directly to the optimization problem [18], although we have preferred to use a simple implementation of Newton's method with constant step continuation, just for the sake of a rapid comparison.

4. Parallel Pareto manifold calculation

Although the reduction to a TPBVP is not an attractive proposition, applying ideas from the numerical solution of such problems by finite differences [17,11], we can develop a novel continuation approach for convex problems that will deform an initial discrete guess of a convex Pareto manifold into its actual target, maintaining an equidistant discretization. One of the most attractive features of such an approach is parallelization, which is not possible with regular continuation from one initial solution, an essentially sequential process. A different approach with apparently similar amount of work can be found in [2].

Going back to problem (1) and the resulting system of optimality conditions (2), we consider the initial mesh of approximate solutions $x_i^{(0)}(\lambda_i)$ defined in (4). Instead of the standard sequential continuation approach mentioned above we consider now deforming the whole set of solutions until it matches the Pareto manifold with enough accuracy.

One desirable feature, emphasized in algorithms such as those of [3,2], is to have an equidistant representation of the Pareto manifold. Observe that we start with such a mesh, but in order to guarantee that we end with an approximately uniform representation we will also solve for the parametrization λ_i , insisting that the spacing (essentially arc length along the manifold) stays uniform, by imposing the additional conditions:

$$L_i(x, \lambda) = \|x_i - x_{i-1}\|_2^2 - \frac{P}{l+1} = 0,$$

where $P = \sum \|x_j - x_{j-1}\|_2^2$. Some authors include this kind of approach into the linear aggregation methods, meaning simply that the parameter λ_i is not prescribed but is calculated along with the solution x_i .

Although this global approach seems to lead to a formidable system of nonlinear equations, it is a very structured one and careful crafting of the solution algorithm results in a reasonable floating point operation count.

Applying Newton's linearization to the system (2)–(4) we obtain,

$$\mathbf{E} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} -\mathbf{G}_1 \\ -\mathbf{G}_2 \\ \vdots \\ -\mathbf{G}_n \\ -\delta \mathbf{x}_0 + \frac{P}{(l+1)^2} \\ \vdots \\ -\delta \mathbf{x}_{l-1} + \frac{P}{(l+1)^2} \end{bmatrix}, \tag{5}$$

where $\delta \mathbf{x}_i = \|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2^2$, and

$$E = \begin{bmatrix} \mathbf{G}_{\mathbf{x}_1} & 0 & 0 & \mathbf{G}_{\lambda_1} & 0 & 0 \\ 0 & \mathbf{G}_{\mathbf{x}_2} & 0 & 0 & \mathbf{G}_{\lambda_2} & 0 \\ 0 & & \mathbf{G}_{\mathbf{x}_l} & & & \mathbf{G}_{\lambda_l} \\ \mathbf{L}_{l\mathbf{x}_1} & & \mathbf{L}_{l\mathbf{x}_l} & \mathbf{0} & & \mathbf{0} \\ \mathbf{L}_{l\lambda_1} & & \mathbf{L}_{l\lambda_l} & \mathbf{0} & & \mathbf{0} \end{bmatrix}.$$

Or in block matrix form:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix},$$

where $A = \text{diag}(G_{\mathbf{x}_i})$, $B = \text{diag}(G_{\lambda_i})$, and \mathbf{a} , \mathbf{b} are the corresponding right hand side vectors from (5).

This 2×2 block system has a special structure that we will exploit. The lower block is full and of size $l \times nl$. The upper blocks are diagonal with a total of $l \times l$ blocks each. The first one has blocks of size $n \times n$, while the second one has blocks of size $n \times 1$, for a whole matrix of size $(n + 1)l \times (n + 1)l$. The partial *Frechet* derivatives corresponding to the L-equations are:

$$\frac{\partial \mathbf{L}_m}{\partial \mathbf{x}_i} = \begin{bmatrix} 2(x_i - x_{i-1}) + \eta_i, & i = m \\ -2(x_i - x_{i-1}) + \eta_i, & i = m - 1 \\ \eta_i, & \text{all other } i's \end{bmatrix},$$

where

$$\eta_i = -\frac{2}{l + 1}(2x_i - x_{i-1} - x_{i+1}).$$

We use block Gaussian elimination to solve this 2×2 block system.

We first eliminate \mathbf{C} by multiplying the first row by $\mathbf{C} \text{diag}(\mathbf{G}_{\mathbf{x}_i}^{-1}) = \mathbf{C} \mathbf{G}_{\mathbf{x}}^{-1}$, and replacing the second row by its difference with the modified first row, thus obtaining a 0 matrix in the low left corner and the Schur complement in the right one:

$$\mathbf{D}^* = \mathbf{C} \mathbf{G}_{\mathbf{x}}^{-1} \mathbf{G}_{\lambda},$$

$$\mathbf{b}^* = -\mathbf{C} \mathbf{G}_{\mathbf{x}}^{-1} \mathbf{G} - \mathbf{b}.$$

Then, back-substituting:

$$\Delta \lambda = \mathbf{D}^{*-1} \mathbf{b}^*,$$

and finally:

$$\Delta \mathbf{x}_i = -\mathbf{G}_{\mathbf{x}_i}^{-1} \mathbf{G}_{\lambda} \Delta \lambda_i - \mathbf{G}_{\mathbf{x}_i}^{-1} \mathbf{G}_i.$$

Since all these last systems are independent, they can be solved in parallel. In fact, for each i we factorize $\mathbf{G}_{\mathbf{x}_i}$ and then solve the two systems:

$$\mathbf{G}_{\mathbf{x}_i} \mathbf{c}_i = \mathbf{G}_i, \quad \mathbf{G}_{\mathbf{x}_i} \mathbf{d}_i = \mathbf{G}_{\lambda_i},$$

to obtain:

$$\Delta \mathbf{x}_i = -\mathbf{d}_i \Delta \lambda_i + \mathbf{c}_i.$$

With these corrections, we step the solutions:

$$\begin{aligned} \mathbf{x}_i^{(j+1)} &= \mathbf{x}_i^{(j)} + \Delta \mathbf{x}_i, \\ \lambda_i^{(j+1)} &= \lambda_i^{(j)} + \Delta \lambda_i. \end{aligned}$$

It is wise to control the Newton step to increase its convergence region. In particular we should insist in the monotonicity of the parameters λ_i , which can also be enforced by controlling the length of the Newton step.

We observe that Gaussian elimination applied directly to the full system, ignoring its structure, would require $(n + 1)^3 l^3 / 3$ operations, while the proposed algorithm has complexity $(2l + 1)n^3 / 3 + l^3 / 3$, which is substantially simpler for large l (the number of Pareto points). The parallel algorithm also speeds up with increasing l .

4.1. An uniform discrete representation of the Pareto front

In the previous section we showed how to obtain a uniform discrete representation of the Pareto manifold, i.e. a set of equally spaced Pareto equilibrium points in design parameter space. However, it is usually simpler to inspect and make decisions by looking at the Pareto front, which is the image of the Pareto manifold in objective space. Unfortunately,

equal spacing in design space does not guarantee equal spacing in objective space. If that is the goal, then it needs to be stated explicitly by replacing conditions (4) by:

$$L_i = \|\mathbf{F}(\mathbf{x}_i) - \mathbf{F}(\mathbf{x}_{i-1})\|_2^2 - \frac{\sum_{k=1}^{l+1} \|\mathbf{F}(\mathbf{x}_k) - \mathbf{F}(\mathbf{x}_{k-1})\|_2^2}{(l+1)} = 0, \quad i = 1, \dots, l.$$

This of course, requires a re-calculation of the Jacobian of the resulting system (albeit, only its lower block):

$$\frac{\partial L_m}{\partial \mathbf{x}_i} = \left\{ \begin{array}{ll} 2[f_1(x_i) - f_1(x_{i-1})]\nabla f_1(x_i) + 2[f_2(x_i) - f_2(x_{i-1})]\nabla f_2(x) + \eta_i, & i = m \\ -2[f_1(x_{i+1}) - f_1(x_i)]\nabla f_1(x_i) - 2[f_2(x_{i+1}) - f_2(x_i)]\nabla f_2(x) + \eta_i, & i = m - 1 \\ \eta_i & \text{all other } i\text{'s} \end{array} \right\},$$

where

$$\eta_i = -\frac{2}{(l+1)}[2f_1(x_i) - f_1(x_{i-1}) - f_1(x_{i+1})]\nabla f_1(x_i) + [2f_2(x_i) - f_2(x_{i-1}) - 2f_2(x_{i+1})]\nabla f_2(x_i).$$

Finally,

$$b_m = -L_m.$$

5. The general case

We like to point out to the dual aspect of the differential Eq. (1), and similarly to the continuation system of the previous section, with regards to the original problem.

From Table 1, one can see that the complexity of calculating the Pareto manifold as the number of input or design variables increases implies more differential (or algebraic) equations, while an increase in the number of objectives beyond 2 leads to partial differential equations (or continuation with multiple parameters) in spaces of increased dimensionality. Observe that for the bi-objective problem the Pareto manifold is always a curve in the space of input variables parameterized by λ .

The continuation method can be extended to the more than two objectives case although, as mentioned, it will have increased complexity (see [20]). An alternative method is presented in [2].

Our approach can also be extended at the price of introducing more constraints, in order to define an uniformly spaced mesh of solutions in the $(k - 1)$ -dimensional Pareto manifold or in the Pareto front. For instance, for three objectives and a $l \times l$ mesh of Pareto points, we would have to solve the following set of equations:

$$\mathbf{G}(\mathbf{x}(\lambda), \lambda) = \lambda_1 \nabla f_1 + \lambda_2 \nabla f_2 + \lambda_3 \nabla f_3, \quad \lambda_1 + \lambda_2 + \lambda_3 = 1,$$

$$\|\mathbf{x}_{i,j} - \mathbf{x}_{i-1,j}\|_2^2 - \frac{1}{(l+1)^2} \sum_{s=0}^l \sum_{t=0}^l \|\mathbf{x}_{s,t} - \mathbf{x}_{s-1,t}\|_2^2 + \|\mathbf{x}_{s,t} - \mathbf{x}_{s,t-1}\|_2^2 = 0,$$

$$i = 0, \dots, l, \quad j = 0, \dots, l,$$

with similar equations in the j th direction and the distances to points outside the mesh equated to zero.

For the algorithm to obtain an equispaced representation of the Pareto front, the distance constraints should be imposed in objective space.

Table 1

Duality between multi-objective problem (MOP) and differential equation (DE) (or continuation) formulation of the optimality conditions

	MOP	DE/Continuation	Dimension
\mathbf{x}	Input variables	State variables	n
λ	Pareto manifold parameters	Independent variables	l

The method also extends to problems with constraints, by using the complete Lagrangian. It is probably wiser to use well proven nonlinear programming codes to solve each of the individual constrained optimization problems, after effecting the first step of eliminating the λ 's. The equidistance constraints can then be naturally included in the nonlinear program. Parallelization can be achieved by decoupling the problems for different points and keeping a data base of current values of the solutions that is updated asynchronously by each optimization process. One then uses whatever values are present in the data base to calculate the couplings between equations. This will lead to a linearly convergent algorithm which will be protected from individual processor failures (see [15]).

Quasi-Newton approximations can be used if derivative information is not readily available. Direct search methods are also plausible, although we will again be moving in the undesirable direction of large number of evaluations.

In Das and Dennis [2] it is shown the equivalence between their method and those that minimize linear combinations of the objectives, such as the methods that we are presenting here. Thus, a fortiori, we can say with Das and Dennis that these methods are independent of scalings of the objectives, which essentially amount to a change in the parametrization of the Pareto set.

5.1. Examples

We will compare four different methods:

- (1) Global front continuation with equispacing constraints on the Pareto manifold.
- (2) Global front continuation with equispacing constraints on the Pareto front.
- (3) Simple Newton continuation.
- (4) Newton continuation with equispacing constraints.

Method 1 was explained in detail in Section 4. Method 2, a modification of the distance constraints of Method 1 was explained in Section 4.

Method 3 was explained briefly in Section 3. It is used with constant step in λ . It should be able to obtain convex segments of the Pareto manifold, but we do not expect it to sample it uniformly.

Method 4 combines Newton continuation with the front equispacing constraints to calculate simultaneously (x, λ) , so that $\|F(x_i) - F(x_{i-1})\|_2^2 = cte$. The *cte* should be the quotient of the length of the Pareto front divided by the number of points. Since we do not know the length of the Pareto front *a priori*, we guess it to be somewhat larger than the distance between the two end points, attempting to account for the curvature of the front and the parameterization mapping.

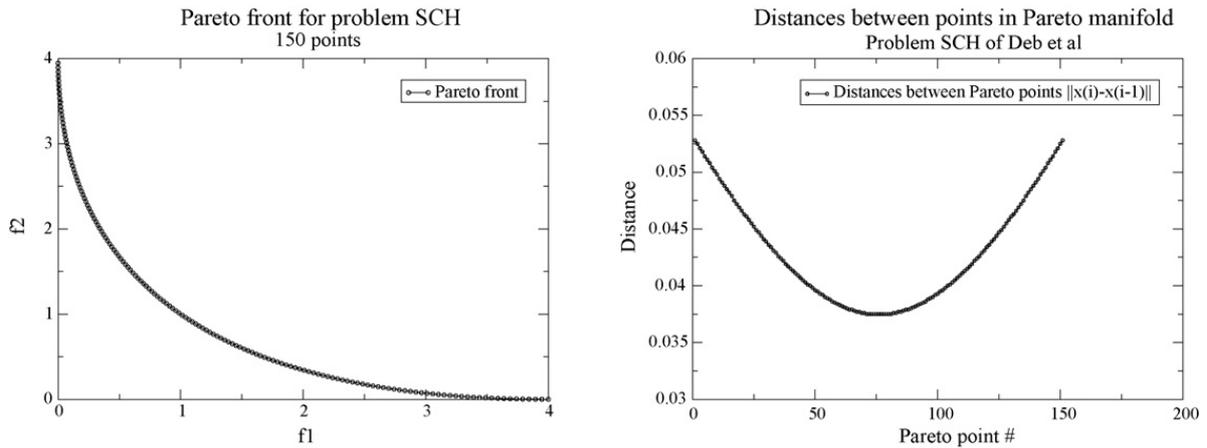
We consider the first test problem (SCH) in [3].

$$\begin{aligned} f_1(x) &= x^2, \\ f_2(x) &= (x - 2)^2. \end{aligned}$$

This problem is obviously trivial. The unique minima of the goal functionals are: $x_1^* = 0$, $x_2^* = 2$, and the Pareto manifold is the whole interval $[0, 2]$. Thus, the problem we need to solve (2)–(4), has $2 \times l$ equations and $2 \times l$ unknowns, in order to generate l points equally spaced in the Pareto manifold. We stop when the RMS is less than $10^{-6}/\sqrt{(2 \times l)}$. Also, the mapping from λ to $x(\lambda)$ is trivial and preserves equidistance: i.e. a uniform mesh in λ will result on a uniform spacing both on input and goal space, so all methods work well, as we will see below.

Since taking the initial values (4) would lead in this case to the solution of the problem, we perturb them so that x and λ are chosen at random within 0.5 of those values.

For most problems, Newton's method takes a maximum of four iterations to reduce RMS below 10^{-6} . Since each iteration requires one evaluation of the vector goal functional and its first and second derivatives (all trivial in this problem), we can say in the language of evolution algorithms that the method requires four generations of a population with l points, to converge to high precision, in contrast with evolutionary algorithms that usually require hundreds of generations to converge to the Pareto front (see [3]).



Results for problem SCH of Deb et al [3]. Simple continuation with equispacing in the Pareto set.

Observe that, as expected and anticipated, for the method described in Section 4, the points in the Pareto front are not equispaced, although they are equispaced in the Pareto set. However, in this simple problem the Pareto front sampling is quite satisfactory. The following figure shows the results with the equispaced constraints on the Pareto front (30 points), which of course are perfectly spaced by construction (Fig. 2).

It is interesting to contrast these results on a trivial problem with the evolutionary methods of [3] as shown below. We see that the evolutionary methods, specially the most recent algorithm NSGA-II, which are extremely costly in terms of function evaluations, do a good job in sampling the front, although they are not perfectly evenly spaced as we saw above and leave a number of gaps in the coverage, besides of requiring several orders of magnitude more functional evaluations.

According to [3], the results for NSGA-II in Fig. 3 were obtained using a population of 100 points and a maximum of 250 generations, or 25,000 function evaluations! They are compared with an earlier method (PAES) to show its improved sampling capability.

It turns out that Method 4, although based on linear aggregates of the objectives, is capable of obtaining points in the non-convex part of the Pareto front, as we see in the next non-convex example (FON), also from [3](Fig. 4):

$$f_1(x) = 1 - e^{-\sum_{i=1}^3 (x_i - 1/\sqrt{3})^2},$$

$$f_2(x) = 1 - e^{-\sum_{i=1}^3 (x_i + 1/\sqrt{3})^2}.$$

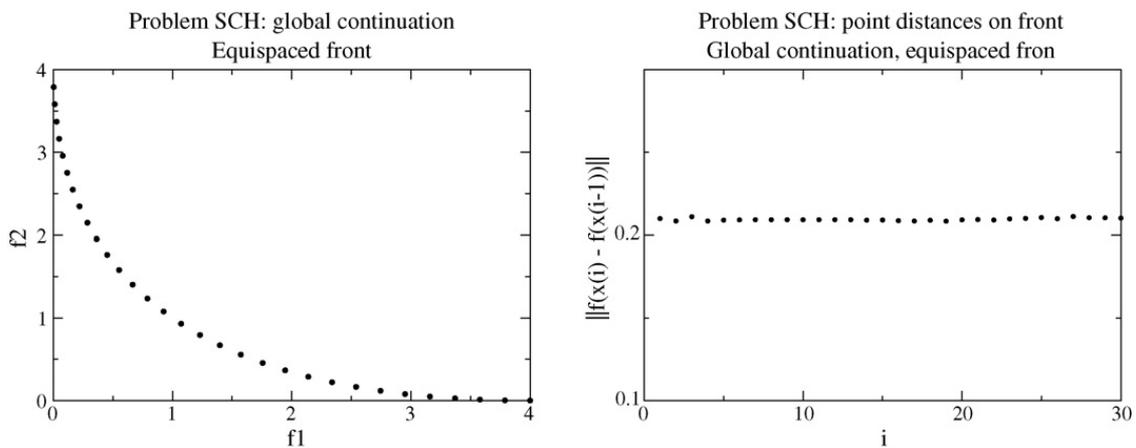


Fig. 2. Problem SCH with global continuation.

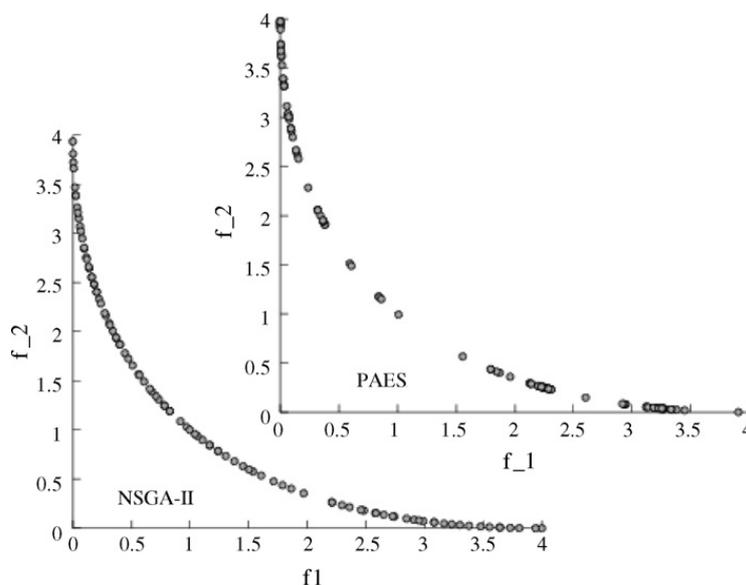


Fig. 3. Results for SCH from [3].

Observe that the simple continuation not only produces irregularly spaced Pareto points, but also only covers a small part of the initial convex segment (beware of the different scales in the two graphs!), while Method 4 has been able to continue through the inflection point and into the concave part of the front, besides of producing a nicely equispaced sample. This seems to contradict the results of [14], and therefore, it requires a closer inspection. We are missing one point at the end since we did not estimate the length of the Pareto front accurately enough.

A more challenging problem is POL, also from [3], since it is non-convex and disconnected (Fig. 5):

$$f_1(x) = 1 + (A_1 - B_1)^2 + (A_2 - B_2)^2,$$

$$f_2(x) = (x_1 + 3)^2 + (x_2 + 1)^2,$$

where,

$$A_1 = 0.5 \sin 1 - 2 \cos 1 + 2 \sin 2 - 1.5 \cos 2,$$

$$A_2 = 1.5 \sin 1 - \cos 1 + 2 \sin 2 - 0.5 \cos 2,$$

$$B_1 = 0.5 \sin x_1 - 2 \cos x_1 + \sin x_2 - 1.5 \cos x_2,$$

$$B_2 = 1.5 \sin x_1 - \cos x_1 + 2 \sin x_2 - 0.5 \cos x_2.$$

Here we see that the methods correctly stop when reaching the end of the left segment and again, plain Newton does a poor job with the spacing while Method 4 gives a nicely sampled front segment.

Now we consider a simplified version of Example 1 of [20]:

$$f_1(x) = (x_2 - 1)^2 + (x_3 - 1)^2 + (x_1 - 1)^4,$$

$$f_2(x) = (x_1 + 1)^2 + (x_3 + 1)^2 + (x_2 + 1)^4.$$

This is a convex problem, but as we see below, the Pareto front has high curvature, thus the global continuation procedure equispaced in input space (Pareto manifold) does not produce an equispaced Pareto front, while the shooting method with equispacing condition on the front does. However, we still have a problem in getting the last points in the front, although we have tried to adjust the step.

Unfortunately, Method 2 that would be the solution to these two difficulties, does not seem to work for this problem, when starting from the simple guess of equispaced points on the straight line joining the minima of the two objectives. However, if we use as initial guess the values obtained by the marching continuation (Method 4), we can successfully calculate an equispaced representation of the Pareto front (Fig. 6).

Finally, we consider a problem with 10 input variables to see what is the impact on performance. We chose initially problem ZDT4 from [3], but this problem is non-convex and $f_1(\mathbf{x}) = x_1$ has a minimum at $-\infty$. The problem specifies

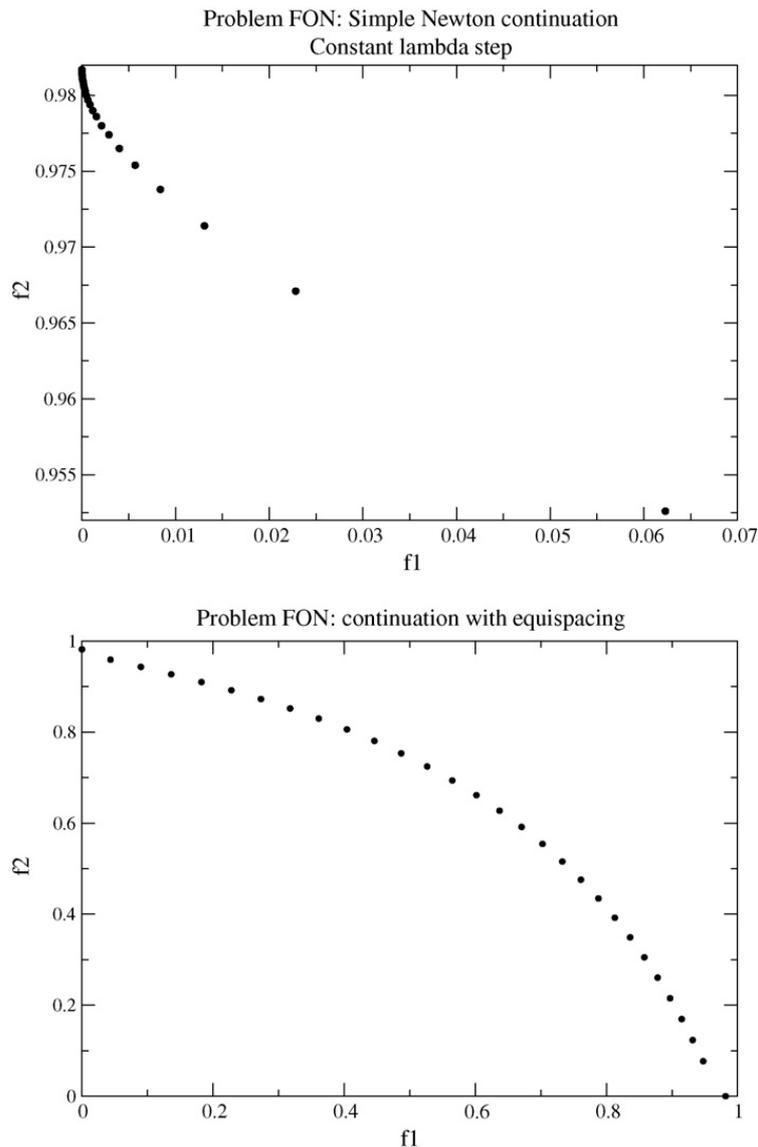


Fig. 4. Problem FON. Both methods used 30 points.

x_1 in $[0, 1]$, which makes it essentially a bound constrained problem, which is outside the scope of this paper. Thus, we changed f_1 to $f_1(x) = x_1^2$, making the problem convex and with a minimum at $\mathbf{x} = 0$. To complete the specification of the problem:

$$f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}], \quad g(x) = 1 + 10(n - 1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)], \quad n = 10.$$

In Fig. 7, we see the results for simple continuation and for continuation with equispaced front constraints and in Table 2 we see that there is no significant impact by the increase in dimensionality. Again, simple continuation does not do a good job, but continuation with equispaced constraints produces an excellent representation of the Pareto front in about 120 function evaluations (for 30 points in the front). The RMS upon convergence is never greater than 10^{-12} (this combines the norm of the gradient with the residual of the equispacing constraints). Although they are not directly comparable, Deb et al. took 30,000 function evaluations for a fairly poor approximation to the front in the unmodified problem ZDT4, while needing 50,000 evaluations to obtain a reasonable approximation.

In terms of performance we offer some timings in Table 2. For these simple problems the times are essentially negligible. We emphasize that since they are all based on Newton's method they use very few function evaluations as

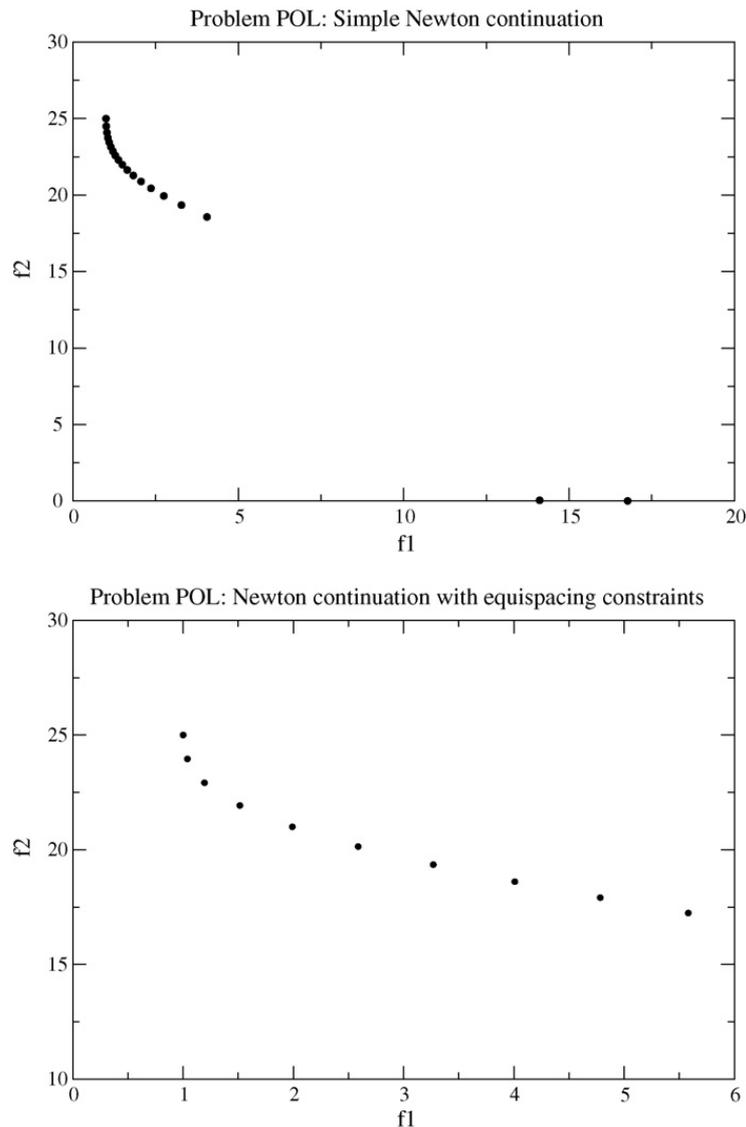


Fig. 5. Results for problem POL (30 points).

compared to evolutionary algorithms. This will be a distinct advantage when applying these methods to problems with expensive function evaluations, provided the necessary derivatives are available, hardly the case in problems where large scale modeling is required to evaluate the goals and constraints, but it might be useful if those are replaced by differentiable surrogates [1,7,16]. The time for Method 2 in problem SCHUTZE1 includes the initialization time by Method 4.

Table 2
Performance on an Intel dual Xeon 3.2 GH machine (LINUX) (time in ms)

Method	Problem				
	SCH	SCHUTZE1	FON	POL	ZDT4 (mod.)
1	2	23	–	–	
2	6	10	–	–	
3	4	–	4	39	8
4	6	7	15	8	20

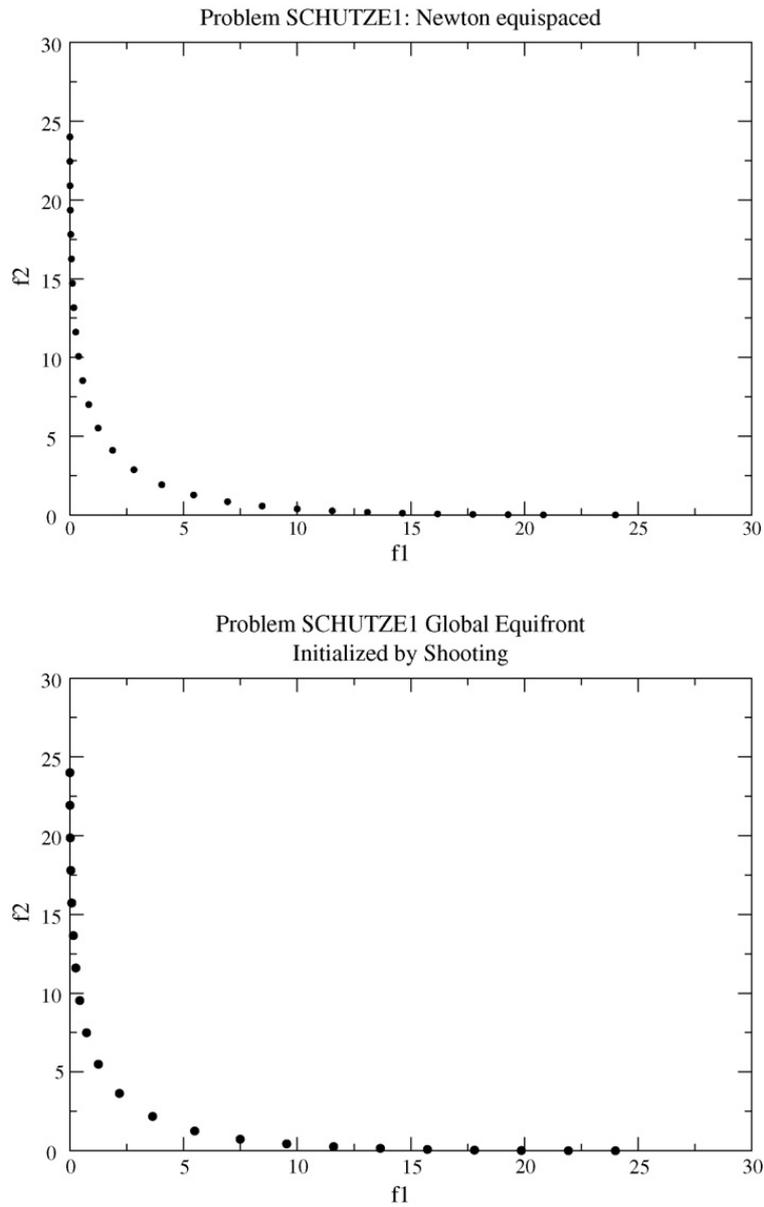


Fig. 6. Results for problem SCHUTZE1 (30 points).

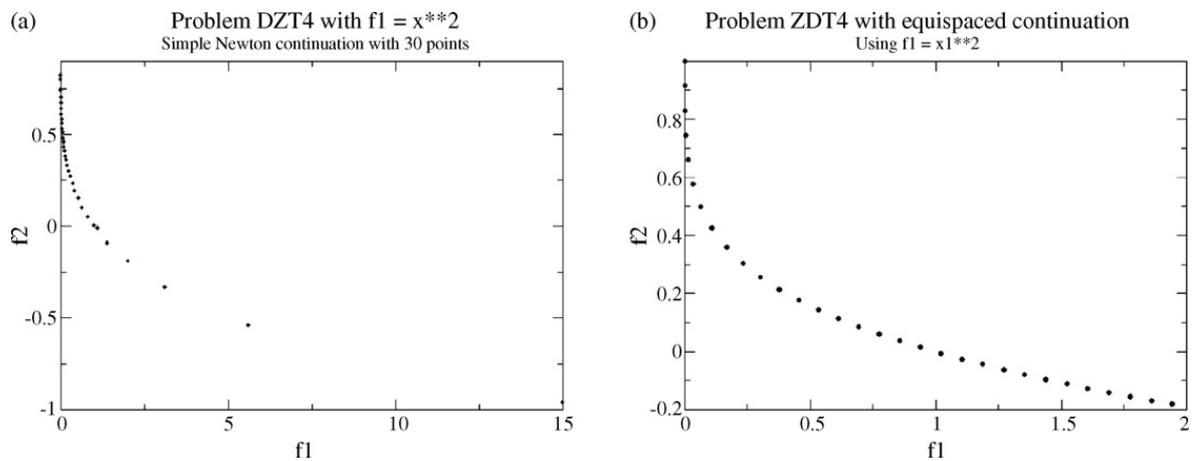


Fig. 7. Results for modified ZDT4 problem.

6. Conclusions

We have discussed in detail the problem of finding well sampled representations of the Pareto manifold or of its image by the objectives, the Pareto front, in a unconstrained multi-objective optimization problem. Explicit algorithms for bi-objective problems were described in detail, implemented and demonstrated on problems taken from the recent literature. Extensions to more objectives and constrained problems were indicated.

We showed that by adding simple equispacing constraints it was possible to have both simple homotopic continuation of scalarized problems or a novel global "bending" method to produce perfectly evenly spaced representations of either the Pareto manifold or the Pareto front.

The algorithms, based on the solution of the KKT first order optimality conditions by Newton's method were shown to use two orders of magnitude less objective evaluations than commonly used evolutionary algorithms. Although we agree with the fact that evolutionary algorithms are applicable to a wider class of problems, we like to emphasize that for problems that are common in Engineering and Scientific applications involving expensive simulations in order to evaluate the objectives and constraints, the proposed methods could be a useful alternative.

Acknowledgements

We like to thank John Dennis and Godela Scherer for valuable comments that helped to improve an early manuscript and the thorough review and excellent suggestions of an unknown referee.

References

- [1] J.J. Alonso, P. LeGresley, V. Pereyra, Aircraft design optimization. *Math. Comp. Simul.*, submitted for publication.
- [2] I. Das, J.E. Dennis, Normal-boundary intersection: an alternate approach for generating Pareto-optimal points in multicriteria optimization problems, *SIAM J. Opt.* 8 (1998) 631–657.
- [3] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. KanGAL Report 200001, Indian Institute of Technology, Kanpur, 2000.
- [4] E. Eschenauer, J. Koski, A. Osyczka, *Multicriteria Design Optimization*, Springer-Verlag, Berlin, 1990.
- [7] G.H. Golub, V. Pereyra, Separable nonlinear least squares: the variable projection method and its applications, *Inverse Prob.* 19 (2003) R1–R26.
- [8] Y. Haimes, W. Hall, H. Freedman, *Multiobjective Optimization in Water Resources Systems*, Elsevier Scientific Publ. Co., Amsterdam, 1975.
- [9] I.Y. Kim, O.L. de Weck, Adaptive weighted-sum method for bi-objective optimization: Pareto front generation, *Struct. Multidisc. Opt.* 29 (2005) 149–158.
- [10] J. Koski, Multicriteria truss optimization, in: W. Stadler (Ed.), *Multicriteria Optimization in Engineering and in the Sciences*, Plenum Press, New York, 1988.
- [11] M. Lentini, V. Pereyra, PASVA4: An ordinary boundary solver for problems with discontinuous interfaces and algebraic parameters, *Math. Appl. Comp.* 2 (1983) 103–118.
- [12] E.R. Lieberman, *Multi-Objective Programming in the USSR*, Academic Press Inc., San Diego, CA, 1991.
- [13] J.G. Lin, Three methods for determining Pareto-optimal solutions of multiple-objective problems, in: Y.C. Ho, S.K. Mitter (Eds.), *Directions in Large-Scale Systems*, Plenum Press, New York, 1975, pp. 117–138.
- [14] A. Messac, G.J. Sundararaj, R. Tappeta, J.E. Renaud, Ability of objective functions to generate points on non-convex Pareto frontiers, *AIAA J.* 38 (2000) 1084–1091.
- [15] V. Pereyra, Asynchronous distributed solution of large scale nonlinear inversion problems. *Appl. Num. Math.*, 30 (1999) 31–40. (special issue, editors: V. Pereyra, J. Castillo).
- [16] V. Pereyra, G. Scherer, F. Wong, Variable projections neural network training. *Math. Comp. Simul.*, 73 (2006) 231–243.
- [17] V. Pereyra, G. Sewell, Mesh selection for discrete solution of boundary value problems in ordinary differential equations, *Numer. Math.* 23 (1975) 261–268.
- [18] J. Rakowska, R.T. Haftka, L.T. Watson, Tracing the Efficient Curve for Multi-Objective Control-Structure Optimization. TR 91–2, Virginia Polytechnic, Blacksburg, 1991.
- [19] W.C. Rheinboldt, J. Burkardt, PITCON: University of Pittsburgh Continuation Program, 1991.
- [20] O. Schutze, A. Dell'Aere, M. Dellnitz, On continuation methods for the numerical treatment of multi-objective optimization problems, in: *Dagstuhl Seminar Proceedings 04461. Practical Approaches to Multi-Objective Optimization*, 2005.
- [21] A.A. Schy, D.P. Giesy, Tradeoff studies in multiobjective insensitive design of airplane control systems, in: *AIAA Guidance and Control Conference*, 1981.
- [22] W. Stadler, *Multicriteria Optimization in Engineering and in the Sciences*, Plenum Press, New York, 1988.
- [23] R.B. Stanikov, J.B. Matusov, *Multicriteria Optimization and Engineering*, Chapman & Hall, New York, 1995.