Struct Multidisc Optim (2012) 46:839–852
DOI 10.1007/s00158-012-0797-1

RESEARCH PAPER

On improving normal boundary intersection method for generation of Pareto frontier

S. Siddiqui · S. Azarm · S. A. Gabriel

Received: 17 November 2011 / Revised: 8 March 2012 / Accepted: 29 March 2012 / Published online: 8 May 2012
© Springer-Verlag 2012

Abstract Gradient-based methods, including Normal Boundary Intersection (NBI), for solving multi-objective optimization problems require solving at least one optimization problem for each solution point. These methods can be computationally expensive with an increase in the number of variables and/or constraints of the optimization problem. This paper provides a modification to the original NBI algorithm so that continuous Pareto frontiers are obtained "in one go," i.e., by solving only a single optimization problem. Discontinuous Pareto frontiers require solving a significantly fewer number of optimization problems than the original NBI algorithm. In the proposed method, the optimization problem is solved using a quasi-Newton method whose history of iterates is used to obtain points on the Pareto frontier. The proposed and the original NBI methods have been applied to a collection of 16 test problems, including a welded beam design and a heat exchanger design problem. The results show that the proposed approach significantly reduces the number of function calls when compared to the original NBI algorithm.

S. Siddiqui (✉)
ICF International, 9300 Lee Highway, Fairfax, VA 22031, USA
e-mail: ssiddiqui@icfi.com

S. Azarm
Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, USA

S. A. Gabriel
Department of Civil and Environmental Engineering, University of Maryland, College Park, MD 20742, USA

Keywords Normal Boundary Intersection (NBI) · Multi-objective optimization · Continuous nonlinear optimization · Pareto solutions · Quasi-Newton methods

1 Introduction

Among the most used gradient-based methods for solving multi-objective optimization problems (Steur 1985; Collette and Siarry 2004) are the weighted method and the constraint method (Cohon 2004). However, both these methods often do not produce an even distribution of Pareto points (Das and Dennis 1997) or at times fail to give an accurate representation of the Pareto frontier (Jia and Ierapetritou 2007). The Normal Boundary Intersection (NBI) (Das and Dennis 1998) and its improvements, the Normal Constraint (NC) (Messac et al. 2003) method, are alternatives not affected by these problems and do not sacrifice computational time to better obtain the Pareto frontier. The NBI and NC methods have been improved to solve problems that have a non-convex Pareto frontier (Messac and Mattson 2004).

In this paper, an approach for multi-objective optimization (Marler and Arora 2004) for continuous nonlinear programs is developed by modifying the *original NBI algorithm* (Das and Dennis 1998). The goal is to obtain the Pareto frontier at a much lower computational cost (when compared to the original NBI algorithm) while sacrificing little for accuracy and have a methodology to increase accuracy if needed. The proposed modification will be based on the original NBI algorithm and will modify it so that it is more suitable for engineering design and other nonlinear multi-objective optimization problems.

This paper's approach (hereafter referred to as the *modified NBI method*) has been tested and verified with 16 optimization problem examples. A comprehensive review of the literature was conducted and the main distinctions between the proposed modified NBI method and previous works are as follows:

(i) Traditional gradient-based methods for multi-objective optimization problems (e.g., weighted method, constraint method, NBI, NC) involve solving multiple optimization problems to obtain the Pareto solution points (Cohon 2004), i.e., one optimization problem for obtaining one Pareto point. Not only can this be computationally expensive and tedious, but also lead to repetition of Pareto points so that a good description of the Pareto frontier may not be obtained (Das and Dennis 1998). While NBI and NC were designed to obtain a distinct, evenly spread Pareto frontier (Messac et al. 2003; Messac and Mattson 2004), the methods still involve solving one optimization problem per Pareto point generated. Alternative gradient-based approaches exist which obtain the Pareto frontier without explicitly solving optimization problems. For example, Rakowska et al. (1993) provided an active set algorithm which used homotopy curve-tracking techniques and obtain the Pareto frontier "in one go." The modified NBI method provides a gradient-based algorithm which, in the case of continuous Pareto frontiers for bi-objective optimization problems, obtains the Pareto frontier by solving one optimization problem. For discontinuous Pareto frontiers, the number of optimization problems required is equal to the number of discontinuities (from numerical evidence of the test problems considered).

(ii) While population-based methods such as genetic algorithms (Deb et al. 2002; Goel et al. 2007; Li et al. 2009) and others (Venter and Haftka 2010) can obtain an estimate of the entire Pareto frontier in an "all-at-once" manner, they involve the pre-selection of numerous parameters for those algorithms, which can greatly influence the solution obtained (Isaacs et al. 2008). For example, a genetic algorithm requires the selection of parameters such as the size of population, crossover type and probability, mutation probability, number of generations, etc. Also, due to the stochastic nature of population-based methods, the same solution is often not reproduced, even with the same settings of parameters. Hence, these algorithms are often run many times to obtain a trusted solution set. The method in this paper involves the selection of only a single parameter, which is used to influence the number of Pareto points generated. Moreover, the same results are reproduced every time with the same value of the parameter so the algorithm does not need to be run many times.

(iii) The proposed approach preserves the computational tractability (numerical evidence) of the problem with respect to the number of variables and number of constraints. By contrast, the computational effort for the original NBI algorithm (Das and Dennis 1998) increases at a greater rate with an increase in number of variables, constraints, and number of Pareto points generated for the test problems considered in this paper. It has to be pointed out that while the modified NBI method can be used to solve problems with greater than two objectives, it increases computational time as it would in the case of the original NBI algorithm.

(iv) Gradient-based methods often have trouble generating a Pareto frontier that is nonlinear and/or discontinuous (Cohon 2004; Das and Dennis 1998). Modifications of NBI (e.g., Messac and Mattson 2004; Mueller-Gritschneider et al. 2009) have been developed to handle concave Pareto frontiers (for minimization) and have been shown to work for nonlinear (non-convex) Pareto frontiers as well. This paper provides numerical evidence of generating such Pareto frontiers with reasonable computational effort.

In the following, first, a description of the NBI method will be provided in detail along with the terminology to be used in Section 2. Then, the proposed modification will be described so that NBI can be more efficient in Sections 3 and 4. Simple examples are presented to show the nuance of the modification in Section 5. Finally, additional numerical examples in Section 5 and engineering examples in Sections 6 and 7 are presented to demonstrate a wide range of applications of the approach.

2 Background and terminology

A multi-objective optimization problem in which at least two or more objectives are conflicting is given by

$$\min_{x \in C} F(x) = \left[f_1(x), \ldots, f_n(x) \right]^T, \qquad n \geq 2 \tag{1}$$

where the superscript T represents the transpose function and

$$C = \left\{ x \in \Re^N : h(x) = 0, g(x) \leq 0, a \leq x \leq b \right\},$$
$$F(x) : \Re^N \rightarrow \Re^n, h(x) : \Re^N \rightarrow \Re^{ne}, g(x) : \Re^N \rightarrow \Re^{ni} \tag{2}$$

Assuming that at least two objective functions are conflicting in (2) then no single x^* would generally minimize every f_i simultaneously. A concept of optimality which is useful in the multi-objective framework is that of Pareto optimality as defined below, see, e.g., Das and Dennis (1998).

Dominance and Pareto Optimality The vector $F(\hat{x})$ is said to *dominate* another vector $F(\bar{x})$, if and only if $f_i(\hat{x}) \leq f_i(\bar{x})$ for all $i = 1, \ldots, n$ and $f_j(\hat{x}) < f_j(\bar{x})$ for at least one j. A point $x^* \in C$ is said to be Pareto optimal or an efficient point for (1) if and only if there does not exist $x \in C$ satisfying $F(x) < F(x^*)$. The vector $F(x^*)$ is then called a non-dominated or non-inferior point, and specifically a *Pareto point*. The set of all Pareto points is referred to as the *Pareto frontier.*

Utopia Point The shadow minimum or utopia point is defined (Das and Dennis 1998) as the vector F^* of individual global minima of the objective functions f_i.

The utopia point, hence, can only be achieved if there exists an x^* that is a minimizer for all the individual objective functions. The existence of such an x^* is very rare, and almost all engineering optimization applications have a set of Pareto points. Note that since both the previous and next definition involve a global as opposed to local minimum, their computation requires the use of a global minimizing algorithm. The examples in this paper use a multi-start gradient based solver in MATLAB (MATLAB 2008), which is assumed to be a global minimizer. Also, without loss of generality, it will be assumed from this point on that $F^* = 0$.

CHIM The Convex Hull of Individual Minima (*CHIM*) is the set of all convex combinations of the individual global minima of the objective functions. Specifically, Let x_i^* be the global minimizer of $f_i(x), i = 1, \ldots, n$ over $x \in C$. Let $F_i^* = F(x_i^*), i = 1, \ldots, n$. Let Φ be an $n \times n$ matrix (sometimes known as the *payoff matrix*) whose ith column is F_i^*. Then the set of points in $\Re^n$ that are convex combinations of F_i^*, i.e.,

$$\left\{\Phi\beta : \beta \in \Re^n, \sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0\right\} \tag{3}$$

is referred to as *CHIM*.

The set of feasible vectors in objective space (feasible region of the objective space), $\{F(x) : x \in C\}$ is denoted by Γ, which is a subset of the objective space whose boundary is defined by $\partial\Gamma$. Let the set of Pareto points, the Pareto frontier, be denoted by Π. Let $CHIM_\infty$ be the affine subspace of lowest dimension that contains *CHIM*, i.e., $CHIM_\infty = \left\{\Phi\beta : \beta \in \Re^n, \sum_{i=1}^{n} \beta_i = 1\right\}$. Let $CHIM_+ = \Gamma \cap CHIM_\infty$ or, geometrically, let $CHIM_+$ be an extension of the *CHIM* to $\partial\Gamma$, which informally can be described as the extension of the boundary of the *CHIM* simplex to $\partial\Gamma$. In two dimensions $CHIM_+ = CHIM$. Figure 1 displays this information geometrically for a problem with two

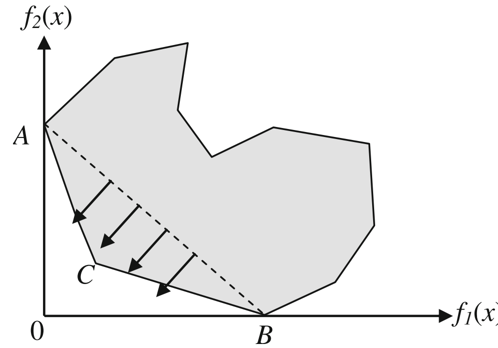

Fig. 1 Description of $CHIM_+$ in the objective space

objective functions. Here, the grey region is Γ, its boundary $\partial\Gamma$, A is F_1^*, B is F_2^*, the line segments ACB is the Pareto frontier Π, and the dashed line segment AB is the $CHIM_+$ (and *CHIM*).

The basic idea behind the original NBI algorithm (Das and Dennis 1998) is to find the Pareto frontier using $CHIM_+$ as a starting point. For different values of β, quasi-normal vectors (as shown in Fig. 1 by arrows) from $CHIM_+$ in the direction of the utopia point are generated. The vectors need not be exactly normal to *CHIM*, but just need to move away from $CHIM_+$ and point towards the utopia point, which is the origin for normalized, minimization problems. Hence, the term quasi-normal is used (Das and Dennis 1998). In particular, the intersection of these vectors with $\partial\Gamma$ will give the Pareto frontier. The optimization problem solved to generate the Pareto frontier is

$$\begin{aligned} &\max_{x \in C,\, t \in \Re_+} \quad t \\ &s.t. \\ &\Phi\beta + t\hat{n} = F(x) \\ &h(x) = 0, g(x) \leq 0, a \leq x \leq b \end{aligned} \tag{4}$$

Here, $\hat{n}$ is the quasi-normal vector emanating from the $CHIM_+$, and t is a real, positive variable. The first constraint in (4) is based on decomposing $F(x)$ into orthogonal vector components. The value of β decides which particular normal vector is being directed towards the Pareto frontier. Moreover, the normal vector $\hat{n}$ does not need to be exact, and in most cases taking $\hat{n} = -\Phi e$ is good enough, as this ensures quasi-normal vectors in the direction of the origin (here, e is the vector of all 1's). The idea is to solve (4) for various values of β to find several points on the boundary of Γ.

Several issues need to be considered when using NBI. First, Γ might not be a convex set, and instead take on complicated shapes that would hinder the method from finding the Pareto frontier. Second, NBI is designed to find the boundary of Γ close to the point of utopia and not the entire

Pareto frontier. In practice, NBI might give points that are on the boundary of Γ, but are not Pareto optimal (Das and Dennis 1998).

Numerous improvements have been made to fix these problems. One major improvement is the Normal Constraint method (Messac et al. 2003) which provides a way to eliminate the non-Pareto points found by the NBI method. Further, some modifications to the Normal Constraint method have been made to deal with a non-convex Γ (Messac and Mattson 2004). Moreover, there is a way to generate Pareto points at the edges of the $CHIM_+$ in multiple dimensions (Mueller-Gritschneider et al. 2009).

Finding the $CHIM_+$ and then solving (4) using different values of β presents a problem that can get computationally expensive. Two ways proposed before (Das 1999; Das and Dennis 1998) to "smartly" parameterize β using a special ordering. Since β is a vector of length equal to the number of objective functions, the ordering involves moving along the first element while keeping others fixed and then the second element and so on (Das and Dennis 1998). This ordering can also be used to solve several problems in parallel, to speed up computation. This essentially entails obtaining starting points that would be close to the next solution and then solving several problems at the same time. For more detailed information, refer to (Das and Dennis 1998).

3 Modified NBI method for solving a bi-objective optimization problem

The first goal of the modified NBI method is to obtain the Pareto frontier by solving a single optimization problem for bi-objective optimization problems with continuous Pareto frontiers. This will be done in two steps. The first step modifies the $CHIM_+$ used in the optimization problem from the one used in the original NBI algorithm. The second step controls the history of iterates of the optimization problem that is solved with this modified $CHIM_+$. To avoid confusion, the term *iterates* will be used to denote iterations of a single optimization problem and not the iterates of an entire algorithm (original NBI algorithm or modified NBI method).

From this paragraph on, our contribution to the modification is discussed. For this section, and the remainder of the modified NBI method, the objective functions need to be normalized so that all objective functions have a minimum at zero and values at the $CHIM_+$ corners to be equal to 1. If the objective function is unbounded or does not attain its maximum, a user-defined upper-bound can be imposed. The normalization is done by dividing each objective function by its maximum function value when the other objective functions are at their individual minimum. An example of this normalization is presented in the numerical results (Section 5). The first step in modifying NBI is using a modified $CHIM_+$ as also considered elsewhere (Mueller-Gritschneider et al. 2009). Equation (5) and Fig. 2 show this change. The $CHIM_{\text{MOD}}$ is at the base, given as a dashed line. Moreover, the problem is changed to find t_{MIN} as opposed to t_{MAX} for the original NBI algorithm:

$$CHIM_{\text{MOD}} = \left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \beta : \beta \in \Re_+, 0 \leq \beta \leq 1 \right\} \tag{5}$$

Hence, the optimization problem to be solved after the modification for a bi-objective optimization problem is

$$\begin{aligned} &\min_{x \in C, t \in \Re_+, \beta \in \Re_+} \quad t \\ &s.t. \\ &\begin{pmatrix} 1 \\ 0 \end{pmatrix} \beta + t \begin{pmatrix} 0 \\ 1 \end{pmatrix} = F(x) \\ &h(x) = 0,\ g(x) \leq 0,\ \beta \geq 0,\ a \leq x \leq b \end{aligned} \tag{6}$$

Note the change in the first constraint. The term $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \beta$ has replaced the term $\Phi\beta$ in (4) because we have replaced $CHIM_+$ with $CHIM_{\text{MOD}}$. Moreover, the vector $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ is normal to the $CHIM_{\text{MOD}}$ and hence is multiplied by t. Compared to the $CHIM_+$ in the original NBI method where the goal was to obtain an even distribution of the Pareto frontier, the goal in the modified NBI method is to use the history of iterates from problem in (6) to obtain the Pareto frontier. Hence, β is included as a decision variable in (6) whereas it was a fixed parameter in (4). Note that (6) is also solved using a gradient-based method.

The next step is to control the history of iterates when solving (6) to obtain the Pareto frontier. An assumption is made that a quasi-Newton method is used to solve (6). Recall the Newton method (Bazaraa et al. 1993) for finding a local minimum for a single-objective optimization

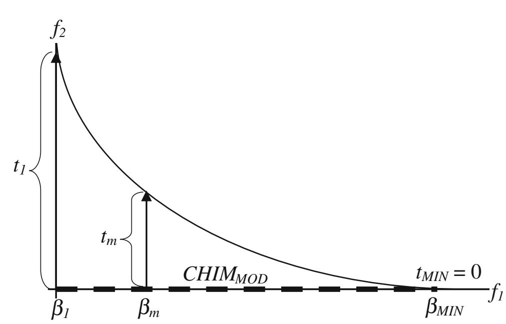

Fig. 2 Modified NBI method, moving along the Pareto frontier from the outside

problem. To find where the first derivative of an objective function is zero (Schoenberg 2001):

$$
\begin{aligned}
&f'(x) = f'(x_m) + f''(x_m)(x - x_m) = 0\\
&x = x_m - \left[f''(x_m)\right]^{-1} f'(x_m)
\end{aligned}
\tag{7}
$$

where the subscript m represents successive better approximations of the solution x^*, i.e., $x_m \to x*$. Under favorable conditions (Bazaraa et al. 1993) convergence is obtained iteratively, where

$$x_{m+1} = x_m - \delta_m \tag{8}$$

$$\delta_m = \left[f''(x_m)\right]^{-1} f'(x_m) = H_m^{-1} g_m \tag{9}$$

Since this is a derivation from the quadratic Taylor series of the objective function, approximating the Hessian H_m can prove troublesome. Hence, another step is added (sometimes referred to as a Newton step) called the line search method in some cases:

$$x_{m+1} = x_m - \alpha_m \delta_m \tag{10}$$

The α_m above in (10) is called the steplength, and δ_m is the direction. However, since (6) is being optimized over three sets of variables, one can rewrite (10) as follows:

$$\begin{pmatrix} x \\ t \\ \beta \end{pmatrix}_{m+1} = \begin{pmatrix} x \\ t \\ \beta \end{pmatrix}_m - \alpha_m \begin{pmatrix} \delta_x \\ \delta_t \\ \delta_\beta \end{pmatrix}_m \tag{11}$$

The driving force behind this is to find t_{MIN} as in Fig. 2, but along the way collect the history of iterations as they will contain important information about the Pareto set. Remember that the original NBI algorithm, (4), chose parameters β. Here, by varying the steplength attached to β, different values of β can be restricted in the history of iterates which can be used to estimate the Pareto set. In other words, (11) is rewritten as (12) to emphasize different steplengths.

$$\begin{pmatrix} x \\ t \\ \beta \end{pmatrix}_{m+1} = \begin{pmatrix} x \\ t \\ \beta \end{pmatrix}_m - \begin{pmatrix} \alpha_m\delta_x \\ \alpha_m\delta_t \\ \alpha_m\delta_\beta \end{pmatrix}_m \tag{12}$$

The goal of this modification is to use the history of successive iterates of a single optimization problem and numerically show that these successive iterates move along the Pareto frontier. To generate a minimum number of points on the Pareto frontier, limiting the magnitude but keeping the same direction of $\begin{pmatrix} \alpha_m\delta_x \\ \alpha_m\delta_t \\ \alpha_m\delta_\beta \end{pmatrix}_m$ will enable the successive approximations of x^* points to have a certain maximum distance in between them. Note that the distance between two successive points on the generated Pareto frontier is given by the size of $\alpha_m\delta_\beta$, which is the amount the variable β "steps" towards the solution. Since it is assumed that all objective functions have been normalized, the magnitude of $\alpha_m\delta_\beta$ can be controlled to generate a desired number of Pareto points. The maximum desired steplength for β at each iteration will be denoted by V_m. For example, if 11 points need to be generated (10 differences) along the Pareto frontier, $\alpha_m\delta_\beta$ can be set to be 0.1 (or 1/10), and if 21 points are needed (20 differences) then $\alpha_m\delta_\beta$ is set to be 0.05 (or 1/20). In general, to generate p points, set $V_m = 1/(p-1)$. Even though V_m is a predetermined constant and is set to be the same for each iteration, the subscript m is included to denote that it is used in each iteration. The new formulation for updating steplength is:

$$\begin{pmatrix} x \\ t \\ \beta \end{pmatrix}_{m+1} = \begin{pmatrix} x \\ t \\ \beta \end{pmatrix}_m - \frac{V_m}{\alpha_m\delta_\beta} \begin{pmatrix} \alpha_m\delta_x \\ \alpha_m\delta_t \\ \alpha_m\delta_\beta \end{pmatrix}_m \tag{13}$$

and V_m will be the quantity varied by the user to help obtain a pre-specified number of Pareto points.

The smaller V_m, the more Pareto points will be generated. V_m needs to be chosen small enough to cover various values of β, as shown in Fig. 3. The remainder of the quasi-Newton method is used in the traditional way (any quasi-Newton method, e.g., BFGS, DFP, Broyden (Bazaraa et al. 1993) will work) to estimate the Hessian matrix and proceed. Realize that one only needs to solve a single optimization problem to obtain the Pareto frontier shown in Fig. 3. A quick convergence without limiting steplength using V_m means fewer Pareto points are generated. As shown in Fig. 3, square points are obtained by choosing a small V_m

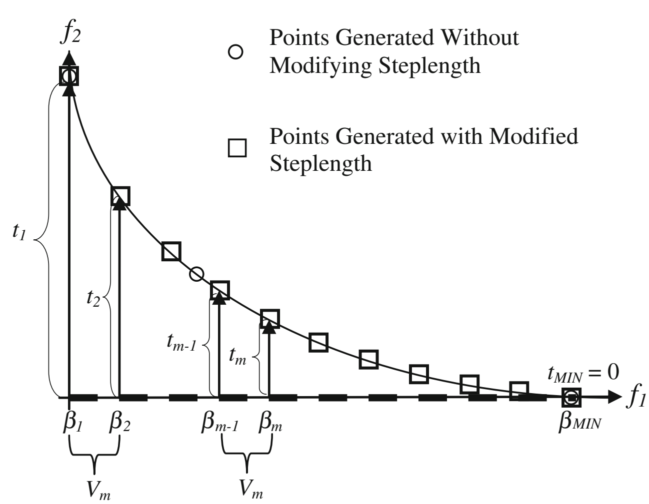

Fig. 3 Pareto points with no restriction on V_m (*circle*) and small V_m (*square*)

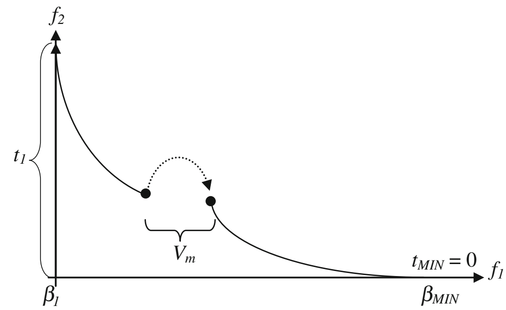

Fig. 4 Choosing the next starting point for an optimization if the solver stops at a point other than $t = 0$

and thus more Pareto points are generated using the history of iterates of the optimization. Circle points denote not limiting V_m, i.e., an application of a quasi-Newton method without limiting steplength. The starting point will be F_2^*, the point at which f_2 is minimized, i.e., when $t = 1$ and $\beta = 0$. By formulation in (6), the objective function t will be minimized generating Pareto points until $t_{\text{MIN}} = 0$ will be reached.

One drawback of the proposed approach is that, if V_m is not chosen to be small, it might give an inaccurate Pareto set due to the fact that each point is not generated by an individual optimization problem but by successive iterates of a quasi-Newton method. Numerical evidence, though, suggests that in the numerical examples considered in this paper the Pareto frontier can be generated with reasonable accuracy by reducing V_m. If computation time is not a limitation, a range of values of V_m can be chosen and the method applied for each value. In any case, all the points found by the proposed NBI modification method will be feasible, which can be confirmed by the solver. Moreover, a Pareto filter, exactly as described elsewhere (Messac et al. 2003) is also applied. This filter ensures the output only displays points that are not Pareto dominated by any other point.

Another issue is that, as with all quasi-Newton methods, discontinuities of the Pareto frontier may cause a suboptimal solution to be obtained, as shown in Fig. 4. The advantage of the modified NBI method is that it is simple to tell when an optimal value of t has been reached. Simply put, the globally optimal value of t in (6), when $CHIM_+$ exists and is finite, will always be 0. Hence, if this value is not reached, the algorithm chooses a starting point V_m distance from the current point away from the starting point as displayed in Fig. 4. If this "jump" is not large enough to counter the discontinuity then the next iteration chooses a starting point of $2V_m$ steps ahead, and so on. This will continue until the optimal value of $t = 0$ is reached.

It is important to note that in cases where the optimization problem stops due to a discontinuity, a new optimization problem with a new starting point (the new point after the V_m jump) is required to generate the remaining Pareto frontier. Hence, in the cases of Pareto frontiers with discontinuities, more than one optimization problem will be required to obtain the Pareto frontier but no more than the number of discontinuities. Table 1 gives a comparison between NBI and the modified NBI method for generating Q points on the Pareto frontier. Note that, as given by Table 1, for problems with continuous Pareto frontiers $D = 0$, one optimization problem is required, which implies that in continuous cases the modified NBI method will computationally perform better than the original NBI algorithm. The number of optimization problems required to solve for a general case is $D+1$, as each discontinuity can at maximum require the solution of one more optimization problem. The cases where it is not expected to computationally perform better than the original NBI algorithm are cases with small I (number of iterates for solving one optimization problem) and large D (number of discontinuities in the Pareto frontier). Note that a smaller I indicates a simpler problem. For the modified NBI method, the maximum number of

Table 1 Comparison between NBI and Modified NBI to Generate Q Pareto Points for a Bi-objective Problem with N Variables, G Constraints, and D Discontinuities along Pareto Frontier

Comparison	NBI	Modified NBI
Number of optimization problems	Q optimization problems with N variables and G constraints	$(D+1)$ optimization problems with $N+2$ variables and $G+1$ constraints
Computational speed	Slow	Quicker
Accuracy	Accurate Pareto set	Approximate Pareto set
Parameters to preset	Have to choose β	Have to choose V_m
Maximum number of function calls given I iterations with M maximum function calls for each iteration	$Q \times I \times M$	$(D+1) \times (max(Q, I)) \times (M+3)$

function calls for each iteration goes up by three because two new variables and one new constraint is added to the optimization problem being solved (compare (4) with (6)). The numerical examples have been selected to have a wide variety of problems to see in which cases the modified NBI algorithm performed better. Another aspect to keep in mind is that Table 1 displays the maximum number of function calls expected, which can be very different than the actual number of function calls observed for a particular problem.

4 Modified NBI method for solving multi-objective optimization problems

The material up until now focused on solving bi-objective optimization problems. However, the modified NBI method can be extended to solving problems with more than two objectives, though it does not preserve the computational tractability as the number of objective functions increases. Consider (1) where there are n objectives.

The first step is to find the $CHIM_+$, and normalize all objective functions. Pick $n - 2$ of the objectives whose objective function value are fixed in a uniform partition of [0, 1]. Then, fixing values of the $n - 2$ objective functions for a specific value in the uniform partition of [0, 1], solve the resulting bi-objective optimization problem. After obtaining a portion of the Pareto frontier, fix to a different set of values within the partition [0, 1]. Hence, problems in which the number of objective functions is greater than two require solving multiple optimization problems, as in the original NBI algorithm. An enumeration on which values to fix is similar to the enumeration provided in (Das and Dennis 1998). A numerical problem with three objective functions is solved in Section 5.2 to display this methodology.

4.1 Algorithm steps

Start

1) Normalize all objective functions to have minimum (point of Utopia) at zero and values at the *CHIM* edges, i.e., maximum values along the Pareto frontier in the objective space to be 1.
2) Fix $n - 2$ objective functions to a fixed value in [0, 1].
3) For each pair, select V_m, the spacing for the generated Pareto points.
4) Initiate the first objective optimization problem with the start point minimum of f_1, $t = 1$, $\beta = 0$.
5) At the end of the optimization, if $t = 0$, $\beta = 1$, then move to step 7 – an approximation to the Pareto set is generated.
6) If not, use $t + V_m$ and $\beta + V_m$ as starting points for the next optimization and return to step 5.
7) If the approximation of the Pareto set needs to be more accurate, use smaller V_m for generation of more Pareto points and return to step 4. Otherwise proceed to step 8.
8) For $n > 2$, use different values in [0,1] as in the original NBI algorithm to fix objective functions values to obtain results for multiple objectives. Then return to step 2. If all combinations of objectives have been fixed, stop. Use the Pareto filter described in (Messac et al. 2003) to remove Pareto dominated points from the generated Pareto set.
9) (*Optional*) For more accurate Pareto set generation, choose a range of V_m and repeat steps 3–8 for all values of V_m in the range. Take the union of all Pareto sets generated after step 8 over different values of V_m and apply the Pareto filter described in (Messac et al. 2003) on this union.

Stop

5 Numerical examples

This section provides numerical examples to support the modified NBI method. First, a step-by-step application to a linear multi-objective optimization problem will be presented. Then, an application to a three-objective optimization problem will be presented to show the extension to multiple objectives. The section will end with a series of numerical tests from the literature which showcase the versatility and speed of the method. Note that original NBI algorithm is used to compare the computational efficiency of all examples. For the CPT 2 example in Section 5.3 and the heat exchanger design in Section 7, no points were generated by the original NBI algorithm. For all examples other than the simple example in Section 5.1, few points were generated with many not on the Pareto frontier for the original NBI algorithm. The authors understand that there might be other problem specific techniques that could have made this application of the original NBI algorithm better, but the goal of this paper is to see how the original NBI algorithm could be improved. Since the focus is on the modified NBI method, the function calls for the original NBI algorithm are still reported where the original NBI method failed to generate points on the Pareto frontier. All single-objective optimization problems in this paper are solved using the *fmincon* solver in MATLAB (2008) with the *active-set* option. One function call is defined as any instance where the solver evaluates an objective or a constraint function.

5.1 Multi-objective linear program simple example (Ex 1 LP)

To demonstrate the modified NBI method, step-by-step, the following linear programming example (Example 1, or Ex 1 LP) is used:

$$\begin{aligned}
&\min\ f_1 = -5x_1 + 2x_2,\ f_2 = x_1 - 4x_2\\
&s.t.\\
&\quad g_1(x) = -x_1 + x_2 \le 3\\
&\quad g_2(x) = x_1 \le 6\\
&\quad g_3(x) = x_1 + x_2 \le 8\\
&\quad g_4(x) = x_2 \le 4\\
&\quad g_5(x) = -x_1 \le 0\\
&\quad g_6(x) = -x_2 \le 0
\end{aligned} \tag{14}$$

The first step in the algorithm is to normalize the objective functions and find the $CHIM_+$. Realize that $F_1^* = [-30\ 6]^T$ and $F_2^* = [3\ \ -15]^T$. The line joining these two points in the objective space is the $CHIM_+$. To normalize f_1 and f_2, replace the objective functions with $f_1^{\text{normalized}} = \frac{f_1-(-30)}{3-(-30)} = \frac{f_1+30}{33}$ and $f_2^{\text{normalized}} = \frac{f_2-(-15)}{6-(-15)} = \frac{f_2+15}{21}$. These functions now have a minimum of 0 and maximum of 1. Next, for this example, the goal is to generate 11 points (11 was chosen for ease of notation. It is completely arbitrary). For all the methods in Table 2, eleven points were aimed to be generated on the Pareto frontier. Hence, select $V_m = 0.1$. The starting point of the algorithm is chosen to be the values of x which minimize f_1, i.e., those that correspond to $t = 1, \beta = 0$. Then, solve the following optimization problem

$$\begin{aligned}
&\min_{x\in\Re^2, t\in\Re_+, \beta\in\Re_+}\ t\\
&s.t.\\
&\quad g_1(x) = -x_1 + x_2 \le 3\\
&\quad g_2(x) = x_1 \le 6\\
&\quad g_3(x) = x_1 + x_2 \le 8\\
&\quad g_4(x) = x_2 \le 4\\
&\quad g_5(x) = -x_1 \le 0\\
&\quad g_6(x) = -x_2 \le 0\\
&\quad \begin{pmatrix}1\\0\end{pmatrix}\beta + t\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}\frac{f_1(x)+30}{33}\\ \frac{f_2(x)+15}{21}\end{pmatrix}
\end{aligned} \tag{15}$$

Starting Point: $t = 1, \beta = 0, x = \begin{bmatrix}6\\0\end{bmatrix}$

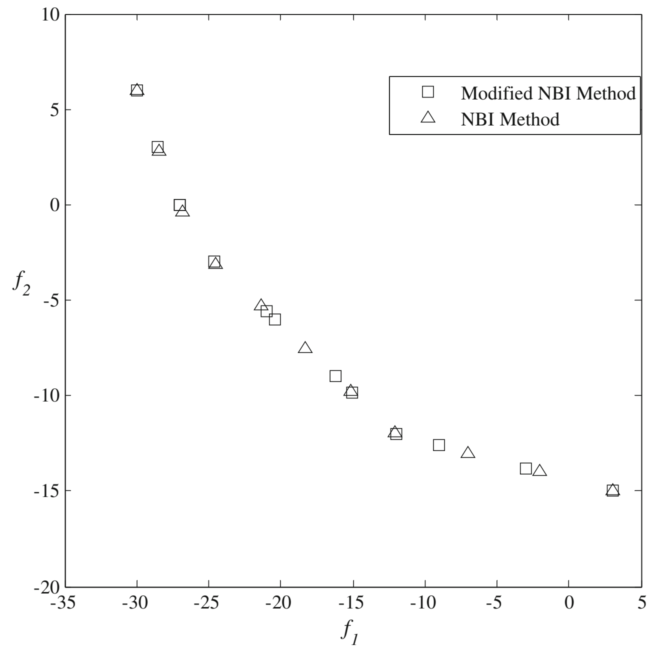

Fig. 5 Solution to Ex1 LP

The history of iterates for solving (15) are the resulting points on the Pareto frontier. The results comparing the NBI methods are shown in Fig. 5. Table 2 displays the results. Four different methods were used to solve this example, which are shown in Table 2. The results show that both the modified NBI method and the original NBI algorithm are able to generate points on the Pareto frontier. The modified NBI method actually generates 12 points, implying that one or more of the steplengths within the iterates was less than V_m. For this example, the original NBI algorithm proves to be the most computationally expensive. The weighted method is not computationally expensive overall, but generates far fewer points on the Pareto frontier. The reason is that the Pareto frontier is made of three linear segments, and the weighted method only generates the four points that connect the three linear segments. The modified NBI method generates the most points with the least number of function calls overall and lowest number of function calls per point generated on the Pareto frontier.

Table 2 Methods used for comparison of Ex 1 LP ($V_m = 0.1$)

Method	Optimization problems	No. of variables	No. of constraints	Function calls	Function calls/no. of points obtained
Weighted method	11	2	6	239	47.8
Constraint method	11	2	7	435	39.5
NBI method	11	3	8	556	50.5
Modified NBI method	1	4	8	216	18.0

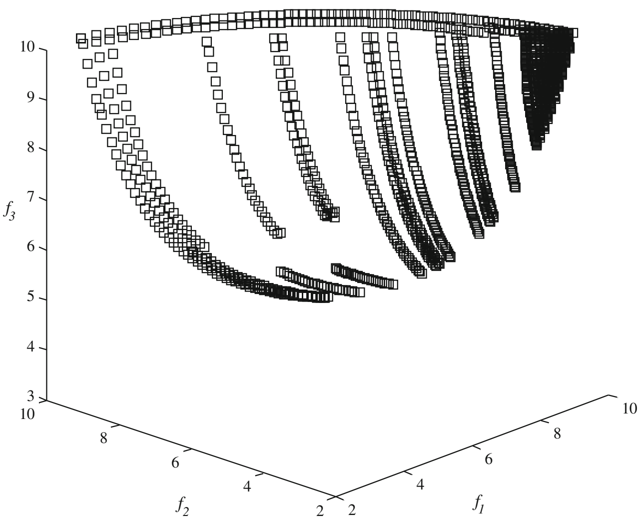

Fig. 6 Solution to test example 2 (Mueller-Gritschneider et al. 2009) using the modified NBI method

5.2 Three-dimensional problem

The Pareto frontier of a three-dimensional problem, referred to as Example 2 (Mueller-Gritschneider et al. 2009) is reproduced below to show the output from the modified NBI method. The first step is to pick one of the objectives which will be fixed. In this case, for example, pick f_2. Then, find the $CHIM_+$; in particular the individual minimum of f_2 will also be found, along with the maximum value that f_2 takes in the $CHIM_+$. In this case, f_2 has a minimum of $f_2 = 3$ and a maximum of $f_2 = 10$. Hence, partition the interval [3, 10] by V_m (0.1 in this example). Then, fixing values of f_2 for each value in the partition of [3, 10], solve the resulting bi-objective optimization problem with f_2 fixed. Hence, objective functions greater than three still require solving multiple optimization problems. However, the results for computational cost are still better than NBI.

Example 2 was solved using 29,546 function calls (456 s on a 2 GHz, 4 GB RAM computer) using the modified NBI method and 3.6×10^8 function calls (20,454 s) using the NBI method. It is important to note that the modified NBI generated a Pareto front similar to the one reported before (Mueller-Gritschneider et al. 2009), which is more accurate than the Pareto front generated from the original NBI method. The Pareto frontier in Fig. 6 is generated in several two dimensional “lines.” This shows that an even generation for multiple dimensions is not possible with the modified NBI method.

5.3 Additional numerical examples

Various other examples were chosen from the literature to show the strength and versatility of the modified NBI method. Problems ZDT 2, ZDT 3, TNK were chosen (Deb et al. 2002) because their Pareto frontiers are “difficult to generate” (Becerra and Coello 2006). Problems DTLZ 9 (Deb et al. 2002) and Example 3 (Messac et al. 2003) were chosen because the constraints of the problems are “difficult to satisfy, hence to generate Pareto optimal solutions” (Becerra and Coello 2006). CTP 2 (Isaacs et al. 2008) was chosen as it is considered a difficult problem with regard to both Pareto frontier and constraints. The following Table 3 shows the results. Note that “Nonlinear” in this table implies functions that are neither convex nor concave. For more difficult problems, a smaller V_m was chosen as to get

Table 3 Comparison of solutions for numerical examples

Test example	Constraints	Variables	Pareto frontier	Step length V_m	Modified NBI function calls	Original NBI function calls	Modified NBI F.Calls/#points	Original NBI F.Calls/#points
Ex 1 LP	6	2	Convex	0.1	396	556	33.00	55.6
Example 3	5	2	Nonlinear	0.01	2,860	6,743	44.7	122.6
ZDT 2	6	2	Concave	0.1	317	762	39.6	69.3
	22	10	Concave	0.1	545	2,204	68.1	200.4
	202	100	Concave	0.1	4,145	20,428	518.1	1,857.1
ZDT 3	6	2	Nonlinear	0.01	4,220	2.11×10^6	29.1	2,144
	22	10	Nonlinear	0.01	33,395	1.66×10^7	230.3	1.65×10^5
	202	100	Nonlinear	0.01	2.66×10^5	1.33×10^9	1,832.1	2.11×10^7
DTLZ 9	5	2	Concave	0.01	2,496	2.49×10^6	46.2	24712
	21	10	Concave	0.01	6,549	5.34×10^7	116.9	5.28×10^5
	201	100	Concave	0.01	51,909	3.24×10^8	910.7	3.21×10^6
TNK	6	2	Nonlinear	0.001	9,065	2.71×10^7	159	27,167
CTP 2	21	10	Nonlinear	0.01	101,145	$> (10^{13})$	1,806.2	–

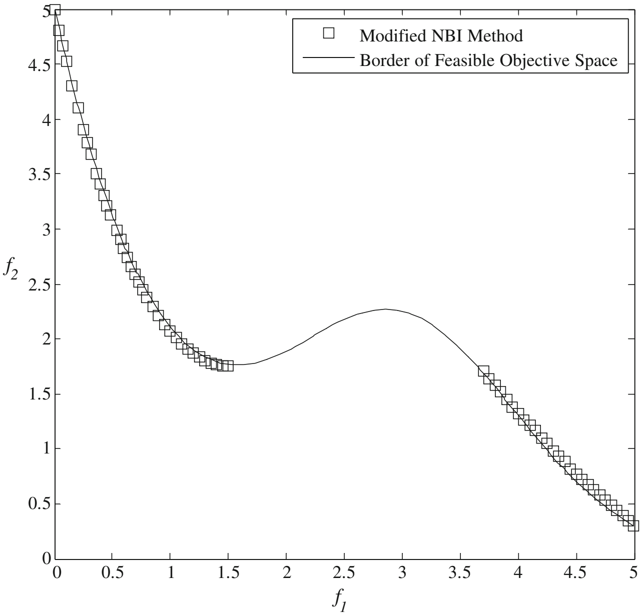

Fig. 7 Solution to example 3

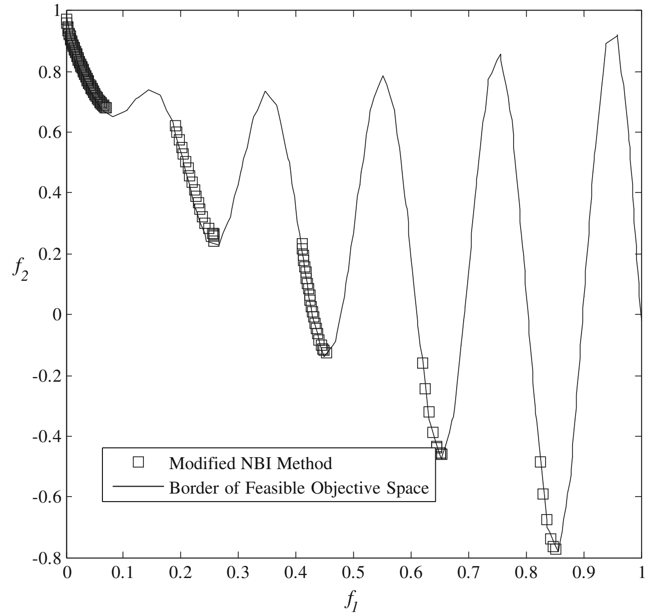

Fig. 9 Solution to ZDT 3

a more accurate representation of the Pareto frontier. This set of examples provides evidence that smaller values of V_m give more accurate representations of a Pareto frontier. CTP 2 could not be solved using the original NBI algorithm for comparison. Hence, a genetic algorithm was used for comparison which could not converge in 10^{13} function calls, so was stopped and no solution was reported.

Table 3 shows that an increase in variables and constraints does not lead to a drastic increase in function calls (and function calls per number of Pareto points generated) in the modified NBI method when compared to the original NBI algorithm. The difference in the number of function calls is from an order of magnitude (for ZDT 2, ZDT 3) to over three orders of magnitude (for DTLZ 9, TNK, CTP 2). Figures 7, 8, 9, 10, 11 and 12 present the Pareto frontiers

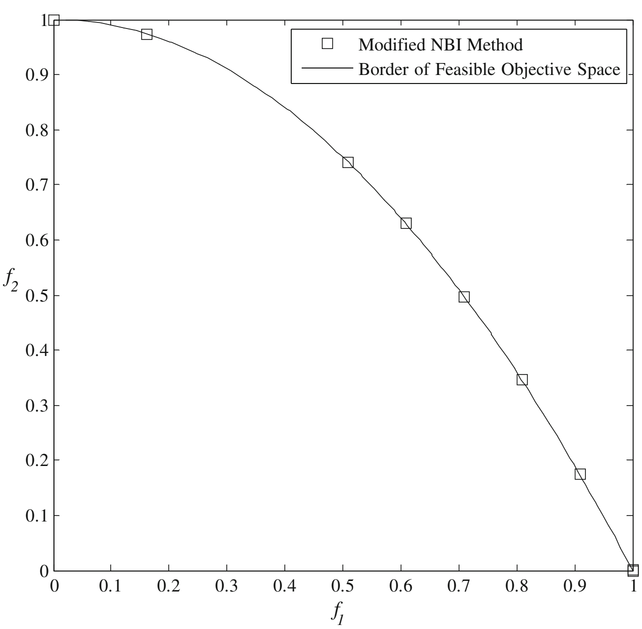

Fig. 8 Solution to ZDT 2

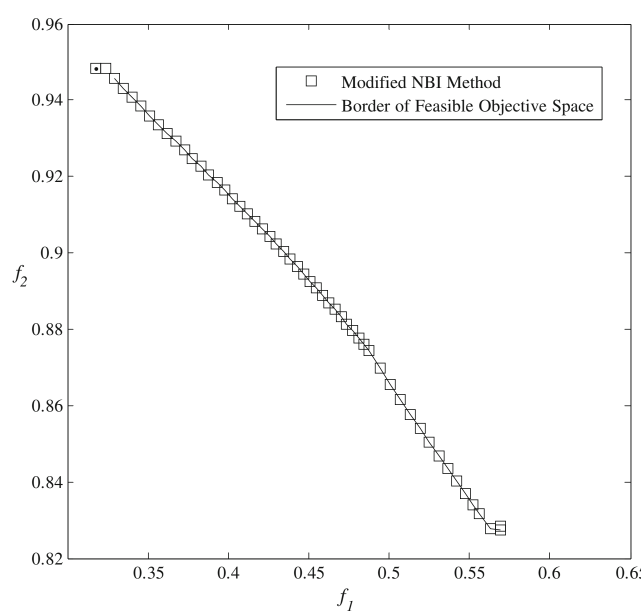

Fig. 10 Solution to DTLZ 9

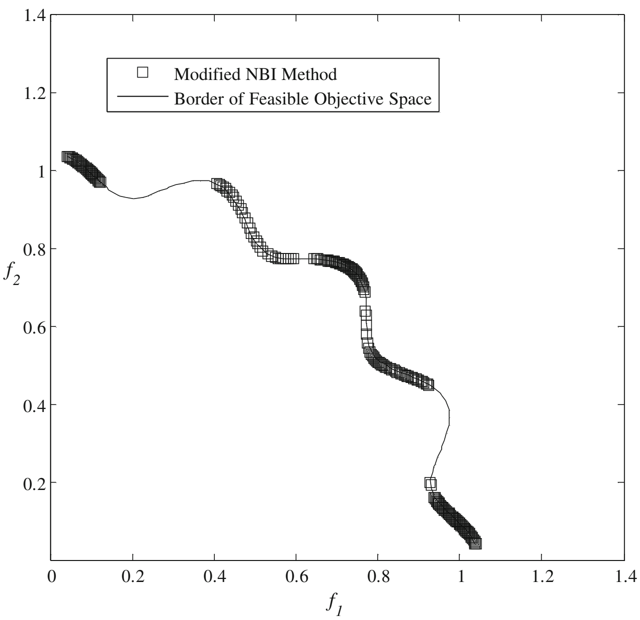

Fig. 11 Solution toTNK

to corroborate the accuracy of the method, with results for ZDT 2, ZDT 3, DTLZ 9 shown for 100 variables.

To show an example of what happens when V_m is not chosen to be large enough, we present the following Fig. 13a and b based on Fig. 7. As given in Table 3, the value of V_m for Example 3 was chosen to be $V_m = 0.01$ and Fig. 7 displays the resulting Pareto set. However, this Pareto set would have been less accurate if V_m was chosen to be

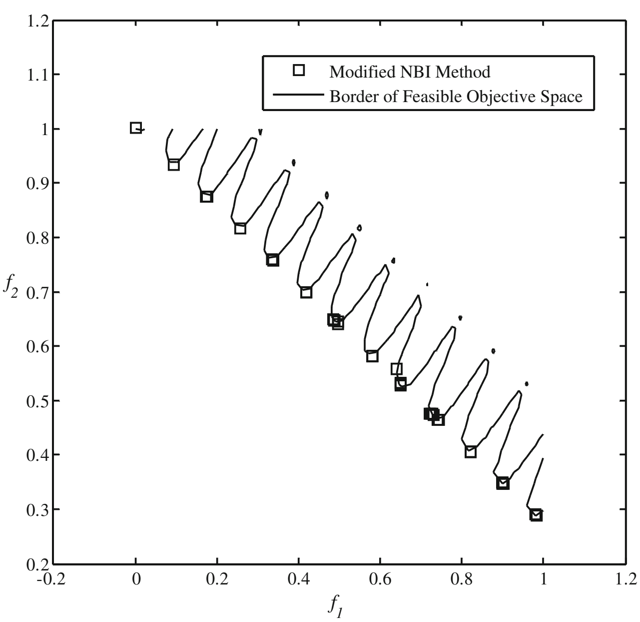

Fig. 12 Solution to CTP 2

$V_m = 0.05$ as shown in Fig. 13a and quite inaccurate if V_m was chosen to be $V_m = 0.1$ as shown in Fig. 13b. Thus, choosing smaller values of V_m or foregoing computational efficiency and using Step 9 in the algorithm in Section 4 can help overcome the problem of inaccurate Pareto set generation under this method.

6 Welded beam design problem (engineering example)

This example is a well-known welded beam problem from the literature (Ragsdell and Phillips 1976). In this problem, a beam is to be welded to a rigid support member. The beam has a rectangular cross-section and is to be made out of steel. The beam is designed to support a force F = 6000 lbf acting at its free end, and there are constraints on the shear stress, normal stress, deflection, and buckling load on the beam. The problem has four continuous design variables, and they are: thickness of the weld, length of the weld, thickness of the beam, and width of the beam. The two objectives of the problem are to minimize the total cost $f_1(x)$ of making such an assembly and the difference between the force F and allowable buckling load $f_2(x)$. For complete formulation of the bi-objective optimization problem including specific values of the parameters, please refer to Gunawan and Azarm (2004).

This example highlights the strength of the modified NBI method over the original NBI algorithm A portion of the Pareto frontier shown by squares is generated that could not be generated by original NBI algorithm. The number of function calls for the modified NBI method (710) was also significantly less than the number of function calls for the original NBI algorithm (35,476) (Fig. 14).

7 Heat exchanger design problem (engineering example)

The following example is from the literature (Magrab et al. 2004) and describes the design of a heat exchanger. The two objectives are to maximize heat transfer (minimize the negative of heat transfer f_2 and minimize the length of the tubing f_1. Several equations govern the heat transfer. The constraints of the model restrict the structure, and the pressure drop on the tube side and shell side. In this example, cold water is in the tubes and hot water is on the shell side and the problem has been setup in a counterflow arrangement for 124 tubes and two-pass heat exchanger. For the complete formulation, please refer to Magrab et al. (2004) and Siddiqui et al. (2011). Figure 15 shows the results generated.

This example could not be solved using the original NBI algorithm. The problem was solved in 18,754 function

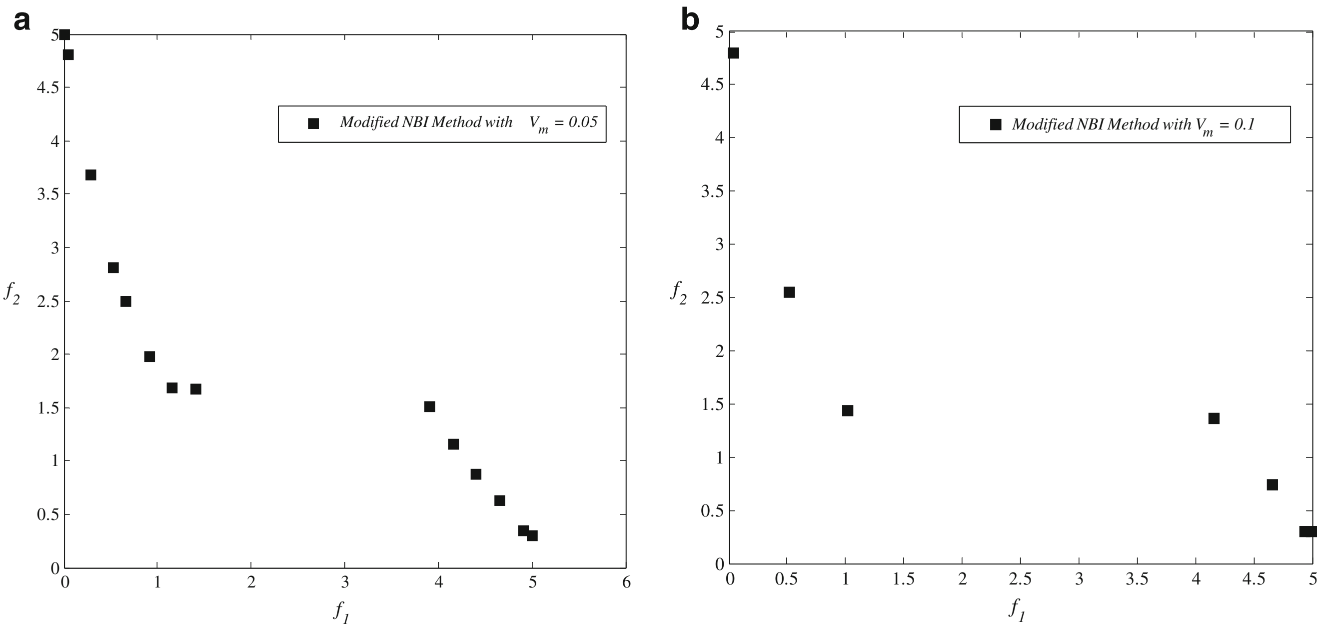

Fig. 13 **a** Ex 3 with $V_m = 0.05$ **b** Ex 3 with $V_m = 0.1$

calls. However, note that the Pareto frontier is generated in clumps. Also, not a lot of points are generated and it took a lot of function calls. This example also shows where the modified NBI algorithm cannot perform so well. The heat exchanger design has a lot of discontinuities in constraint functions and its Pareto frontier. The modified NBI method in this case was not able to provide an evenly distributed accurate Pareto frontier.

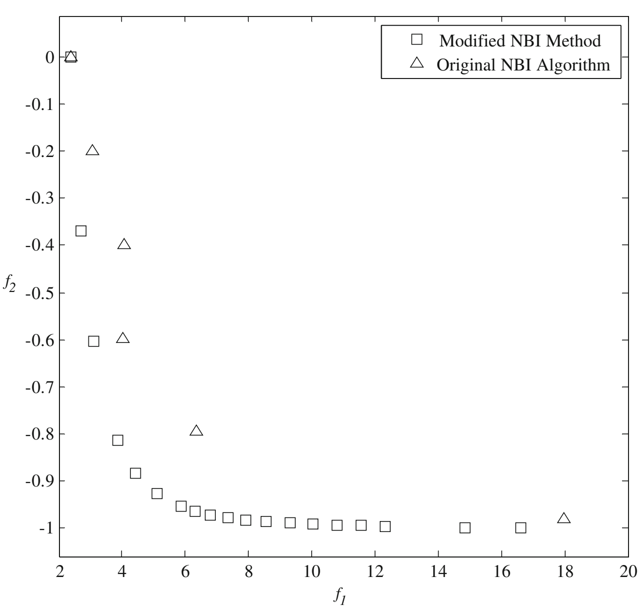

Fig. 14 Results for the welded beam example

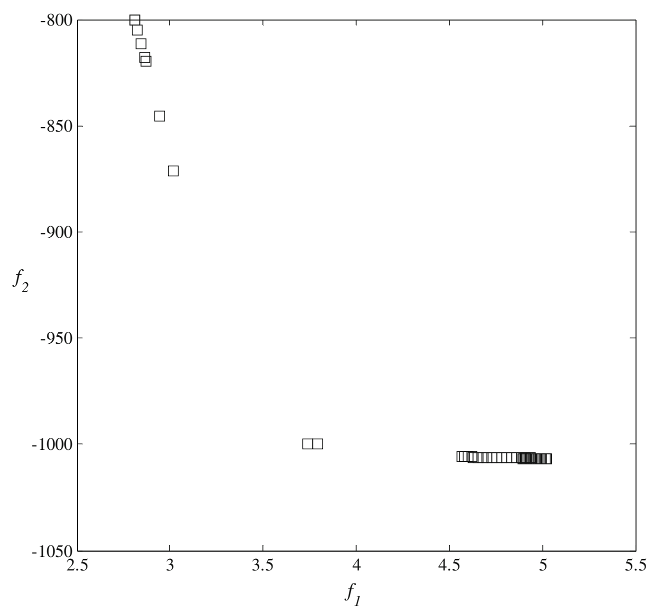

Fig. 15 Heat exchanger design Pareto points as given by the modified NBI method

8 Concluding remarks

This paper presents an improved multi-objective optimization approach to the original NBI algorithm. The proposed modified NBI method obtains Pareto solutions to multi-objective optimization problems with continuous and discontinuous Pareto frontiers with lower computational cost than the original NBI algorithm. The approach is tested with 16 numerical and engineering examples with the most general problem having a nonlinear (non-convex) objective function and nonlinear (non-convex) constraints with discontinuous Pareto frontiers. The modified NBI method provides approximate Pareto solutions to general nonlinear multi-objective optimization problems, with a way to improve this approximation if desired.

The test examples show the strength of the proposed method when compared to the original NBI algorithm. Not only is the method computationally more efficient, but for some examples also obtains better (lower objective function value) solutions when compared to the original NBI algorithm. The number of function calls increases at not an exponential rate with an increase in number of variables and constraints. A wide variety of numerical examples and two engineering examples are presented to verify the approach.

The proposed approach does have limitations. First, for number of objective functions three or greater, the method is not computationally tractable, though it still provides reliable results as in Example 2. Moreover, the method depends on the calculation of the payoff matrix of objective functions. For number of functions greater than three, calculating this matrix can become computationally expensive. Second, as witnessed in the heat exchanger design example, a large number of discontinuities in the problem and/or Pareto frontier can result in an uneven distribution of the Pareto solution points. Third, while this was not evident in the numerical examples, there is no theoretical proof that the modified NBI algorithm will always provide an accurate representation of the Pareto frontier. While the Numerical results are encouraging, as shown in this paper, the theoretical proof is part of future work.

In closing, the main advantage of the modified NBI algorithm is simplicity of the algorithm. It is a gradient-based algorithm, which requires the solution of far less optimization problems than the original NBI algorithm. The other advantage is that it numerically appears to be much more computationally efficient than the original NBI algorithm. Finally, the method is an efficient way to generate multiple points on the Pareto frontier with very low computational effort per point generated.

Acknowledgments The authors wish to thank the anonymous referees and Mr. Amir Mortazavi for their constructive ideas to improve the paper. The work presented in this paper was supported in part by the Office of Naval Research Contract N000140810384. Such support does not constitute an endorsement by the funding agency of the opinions expressed in this paper. The opinions and ideas expressed in this paper are the authors and do not necessarily represent the opinions and ideas of their affiliated institutions.

References

Bazaraa M, Sherali H, Shetty C (1993) Nonlinear programming: theory and algorithms, 2nd edn. Wiley, New York

Becerra R, Coello CAC (2006) Solving hard multi-objective optimization problems using epsilon-constraint with cultured differential equation. In: Runarsson TP et al (eds) PPSN IX LNCS, vol 4193, pp 543–552

Cohon J (2004) Multi-objective programming and planning. Dover Publications, Mineola

Collette Y, Siarry P (2004) Multiobjective optimization: principles and case studies. Springer, New York

Das I (1999) On characterizing the “knee” of the Pareto curve based on normal-boundary intersection. Struct Multidisc Optim 18(2–3):107–115. doi:10.1007/BF01195985

Das I, Dennis J (1997) A closer look at drawbacks of minimizing weighted sums of objectives for Pareto Set Generation in Multicriteria Optimization problems. Struct Optim 14: 63–69

Das I, Dennis J (1998) Normal boundary intersection: a new method for generating Pareto surface in nonlinear multicriteria optimization problems. SIAM J Optim 8(3):631–657

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197

Goel T, Vaidyanathan R, Haftka R, Shyy W, Queipo NV, Tucker K (2007) Response surface approximation of Pareto optimal front in multi-objective optimization. Comput Methods Appl Math Eng 196(4–6):879–893

Gunawan S, Azarm S (2004) Non-gradient-based parameter sensitivity estimation for single-objective robust design optimization. J Mech Des 126(3):395–402

Isaacs A, Ray T, Smith W (2008) Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems. IEEE Congress Evol Comput 2780–2787

Jia Z, Ierapetritou MG (2007) Generate Pareto optimal solutions of scheduling problems using normal boundary intersection technique. Comp Chem Eng 31:268

Li G, Li M, Azarm S, Al Hashimi S, Al Ameri T, Al Qasas N (2009) Improving multi-objective genetic algorithms with adaptive design of experiments and online metamodeling. Struct Multidisc Optim 37(5):447–461

Magrab E, Azarm S, Balachandran B, Duncan J, Herold K, Walsh G (2004) An engineer’s guide to Matlab. Prentice Hall, New York

Marler R, Arora J (2004) Survey of multi-objective optimization methods for engineering. Struct Multidisc Optim 26:369–395

MATLAB (2008) MATLAB and Simulink for technical computing. Mathworks, Version 2008b

Messac A, Ismail-Yahaya A, Mattson CA (2003) The normalized normal constraint method for generating the Pareto frontier. Struct Multidisc Optim 25(2):86–98

Messac A, Mattson CA (2004) Normal constraint method with guarantee of even representation of complete Pareto frontier. AIAA J 42(10):2101–2111

Mueller-Gritschneider D, Graeb H, Schlichtmann U (2009) A successive approach to compute the bounded Pareto front of practical multi-objective problems. SIAM J Optim 20(2):915–934

Ragsdell K, Phillips D (1976) Optimal design of a class of welded structures using geometric programming. J Eng Ind 98(3):1021–1025

Rakowska J, Haftka R, Watson L (1993) Multiobjective control-structure optimization via homotopy methods. SIAM J Optim 3(3):654–667

Schoenberg R (2001) Optimization with the quasi-Newton method. http://www.aptech.com/papers/qnewton.pdf. Aptech Systems Inc., Maple Valley, WA

Steur R (1985) Multicriteria optimization – theory, computation and application. Wiley, New York

Siddiqui S, Azarm S, Gabriel S (2011) A modified benders decomposition method for efficient robust optimization under interval uncertainty. Struct Multidisc Optim 44(2):259–275

Venter G, Haftka R (2010) Constrained particle swarm optimization using a bi-objective formulation. Struct Multidisc Optim 40(1–6):65–76