NORMAL-BOUNDARY INTERSECTION:
AN ALTERNATE METHOD FOR GENERATING PARETO OPTIMAL POINTS
IN MULTICRITERIA OPTIMIZATION PROBLEMS[1]

Indraneel Das and John Dennis
Dept. of Computational & Applied Mathematics
Rice University
Houston, TX 77251-1892

Abstract

This paper proposes an alternate method for finding several Pareto optimal points for a general nonlinear multicriteria optimization problem, aimed at capturing the tradeoff among the various conflicting objectives. It can be rigorously proved that this method is completely independent of the relative scales of the functions and is quite successful in producing an evenly distributed set of points in the Pareto set given an evenly distributed set of 'weights', a property which the popular method of linear combinations lacks. Further, this method can be easily extended in case of more than two objectives while retaining the computational efficiency of continuation-type algorithms, which is an improvement over homotopy techniques for tracing the tradeoff curve.

1 Introduction

A wide variety of problems arising in design optimization of engineering systems are essentially multicriteria in nature (see, for example, Eschenauer, Koski and Osyczka [1] and Statnikov and Matusov [2]). For example, a typical bridge-construction design might involve simultaneously minimizing the total mass of the structure and maximizing its stiffness. However, it is highly improbable that these conflicting objectives would both be 'extremized' by the same design, hence some tradeoff between the objectives functions is desired to ensure an efficient design. Mathematically such a multicriteria optimization problem can be written as:

$$\min_{x \in C} F(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix}, \qquad n \geq 2, \qquad \dots (MOP)$$

where

$$C = \{x : h(x) = 0, g(x) \leq 0, a \leq x \leq b\},$$

$F : \Re^N \mapsto \Re^n$, $h : \Re^N \mapsto \Re^{ne}$ and $g : \Re^N \mapsto \Re^{ni}$ are twice continuously differentiable mappings, and $a \in (\Re \cup \{-\infty\})^N, b \in (\Re \cup \{\infty\})^N$, N being the number of variables, n the number of objectives, ne and ni the number of equality and inequality constraints.

Since no single x^* would generally minimize every f_i simultaneously, a concept of optimality which is useful in the multiobjective framework is that of Pareto optimality, as defined below:

Definition: A point $x^* \in C$ is said to be (globally) Pareto optimal or a (globally) efficient point or a non-dominated or a non-inferior point for (MOP) if and only if $\nexists x \in C$ such that $F(x) \leq F(x^*)$ with at least one strict inequality (the $\leq$ implies term-by-term inequality).

The shadow minimum, F^*, is defined as the vector containing the individual global minima, f_i^*, of the objectives, i.e.,

$$F^* = \begin{bmatrix} f_1^* \\ f_2^* \\ \vdots \\ f_n^* \end{bmatrix}.$$

1

(We assume here and henceforth the existence of a minimum for each of our objectives.) The shadow minimum could thus be attained only in the rare case when a single x minimizes all the objective functions. However, in practical situations, the best we can hope for is to get close to the shadow minimum and assure that there is an agreeable trade-off among the multiple objectives.

Very often in engineering applications the desired solution is a whole collection of Pareto optimal points, representative of the entire spectrum of efficient solutions. Thus ideally, the desired solution is the entire Pareto optimal set, which can be obtained for some small problems which allow themselves to be treated parametrically, resulting in closed-form expressions for the Pareto set (see Lin [3]). More recently, attempts have been made to approximate the entire curve of Pareto optimal solutions in bi-objective problems using techniques which trace the curve of parametrized optima (see Rakowska, Haftka and Watson [4], Rao and Papalambros [5], Lundberg and Poore [6]). The next best solution, which is very acceptable in most applications, is a set of Pareto optimal points obtained by combining the multiple objectives into a single objective function and minimizing the single objective over various values of the parameters used to combine the objectives. For example, it is possible to generate a set of Pareto optimal points by minimizing a convex combination of the objectives, $\alpha^T F(x)$, over $x \in C$, where $\alpha \geq 0$ (component-wise) and $\sum_{i=1}^{n} \alpha_i = 1$, and performing the minimization for different choices of α (see, among many others, Koski [7]). In this article, we propose a new method for generating Pareto optimal points which is at least as efficient as these methods and, unlike the techniques for tracing the curve of Pareto optimal solutions, can be applied to problems with more than two objectives.

2 Preliminaries

First let us introduce some terminology:

Convex Hull of Individual Minima ($CHIM$): Let x_i^* be the respective global minimizers 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 the $n \times n$ matrix whose i^{th} column is $F_i^* - F^*$. Then the set of points in $\Re^n$ that are convex combinations of F_i^*, i.e.,

$\{\Phi w : w \in \Re^n, \sum_{i=1}^{n} w_i = 1, w_i \geq 0\}$, is referred as the *Convex Hull of Individual Minima*.

The set of attainable objective vectors, $\{F = F(x) : x \in C\}$ is denoted by $\mathcal{F}$, so $F : C \mapsto \mathcal{F}$, i.e., C is mapped by F onto $\mathcal{F}$. The space $\Re^n$ which contains $\mathcal{F}$ is usually referred to as the **objective space**. The map of C under F in the objective space is often called the **multi-loss map**[2] (bi-loss map, if $n = 2$). We shall denote the boundary of $\mathcal{F}$ by $\partial \mathcal{F}$. The set of all Pareto optimal points is usually denoted by $\mathcal{P}$. The complete curve/surface of Pareto minima (continuous or not) is often referred to as the **trade-off function** (see p9, Haimes, Hall and Freedman [8]).

$CHIM+$: Let $CHIM\infty$ be the affine subspace of lowest dimension that contains the $CHIM$. Then $CHIM+$ is defined as the smallest simply-connected set that contains every point in the intersection of $\partial \mathcal{F}$ and $CHIM\infty$. More informally, consider extending (or withdrawing) the boundary of the $CHIM$ simplex to touch $\partial \mathcal{F}$, the 'extension' of $CHIM$ thus obtained is defined as $CHIM+$.

Henceforth, it shall be assumed that the objective functions have been defined with the shadow minimum shifted to the origin, so that all the objective functions are non-negative, i.e., $F(x)$ is redefined as:

$$F(x) \leftarrow F(x) - F^*.$$

We observe that in Fig.1, which shows the set $\mathcal{F}$ in the objective space, the point A is F_1^*, B is F_2^*, O is the shadow minimum (and the origin), the broken line segment AB is the $CHIM$, while the 'arc' ACB is the set of all Pareto minima in the objective space; alternately, the trade-off curve. In this (and any) problem with $n = 2$ (i.e., bi-objective), $CHIM = CHIM+$ and the matrix Φ is anti-diagonal.

3 Central Idea

The pivotal idea behind our approach will be introduced by means of a simple observation: the intersection point between the normal emanating from

---

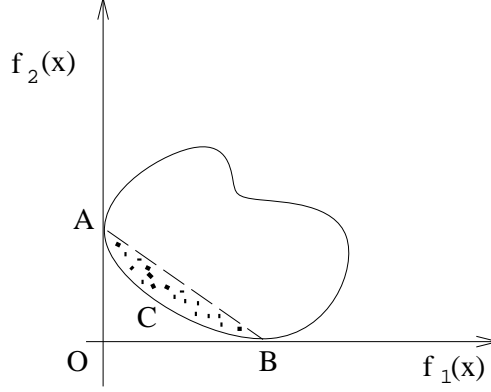[2]This terminology is widely used in game theory.

Figure 1: A typical bi-loss map

any point in the $CHIM$ and the boundary $\partial\mathcal{F}$ is probably a Pareto optimal point; the point of intersection closest to the origin is a Pareto minimal point (while the one furthest is a Pareto maximal point). We say 'probably' because this may not always be true, e.g., when the boundary is 'folded' (see Fig.2). But it is true when the trade-off surface in the objective space is convex, which happens in almost every application found in the literature (see for example the problems in Refs. 1, 2 and 7).

Given a convex weighting w, Φw represents a point in the $CHIM$. Let $\hat{n}$ denote the unit normal to the $CHIM$ simplex pointing *towards* the origin; then $\Phi w + t\hat{n}, t \in \Re$ represents the set of points on that normal. Then the point of intersection between the normal and the boundary of $\mathcal{F}$ closest to the origin is identical to the solution of the following subproblem:

$$\max_{x,t} t$$

$$s.t. \quad \Phi w + t\hat{n} = F(x)$$
$$h(x) = 0 \qquad\qquad (NBI_w)$$
$$g(x) \le 0$$
$$a \le x \le b.$$

The constraints $\Phi w + t\hat{n} = F(x)$ ensure that the point x is actually mapped by F to a point on the normal, while the remaining constraints ensure feasibility of x with respect to the constrained set in the original problem (MOP).
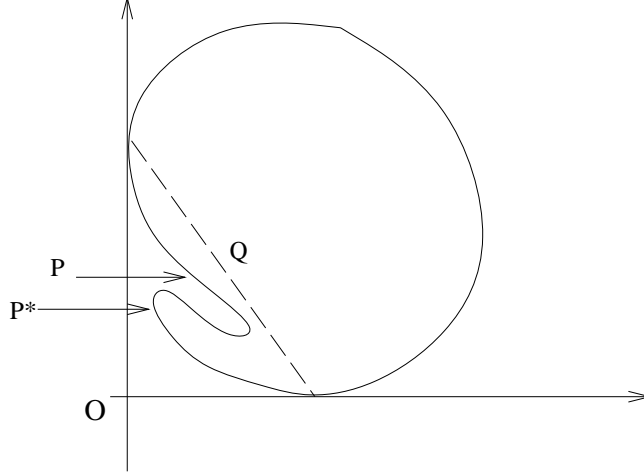
4

Figure 2: NBI started at Q converges to P (locally Pareto optimal), whereas the corresponding globally efficient point would have been P^*.

The subproblem above shall be referred to as the *NBI subproblem*, often written as NBI_w (since w is the characterizing parameter of the subproblem), and solutions of these subproblems will be referred to as *NBI points*. The idea is to solve NBI_w for various w and find several points on the boundary of $\mathcal{F}$, effectively constructing a pointwise approximation to the part of the boundary containing the Pareto minimal set.

As indicated earlier, all NBI points are not Pareto optimal points. For biobjective problems, for every Pareto optimal point there exists a corresponding NBI subproblem of which it is the solution. The same is true for $n \geq 3$, with one difference: the components of the weight w for NBI_w may not add up to 1. As a simple example, suppose $\mathcal{F}$ is a sphere in $\Re^3$ touching the coordinate axes, for simplicity. Then the $CHIM$ simplex is the triangle formed by joining the three points where the sphere touches the axes. Quite clearly, $CHIM \neq CHIM+$ and there exist points in $CHIM + \backslash CHIM$ underneath which there are Pareto optimal points on the sphere. However since these points are not in $CHIM$, they do not satisfy $\sum_1^n w_i = 1$. Thus, by solving NBI_w for $\sum_1^n w_i = 1$, a portion of the Pareto set might be overlooked for problems with $n > 2$. However, these overlooked points are likely to be 'extremal' Pareto points which are not interesting from the tradeoff standpoint, which is our primary goal.

5

3.1 Some details

3.1.1 Structure of Φ

The ith column of Φ is described by

$$\Phi(:,i) = F(x_i^*) - F^*.$$

Since $f_i(x_i^*) = f_i^*$, clearly,

$$\Phi(i,i) = 0.$$

Furthermore, if x_i^* is the global minimizer of $f_i(x)$, then

$$\Phi(j,i) \geq 0, j \neq i.$$

Thus, a negative element in position (j,k) of Φ signifies that x_k^* is not the global minimizer of $f_k(x)$, and $f_k(x_j^*) < f_k(x_k^*)$, i.e., x_j^* improves on the current local minimum of $f_k(x)$. This very fortunate occurrence can help refine the local minimum of an objective by a simple examination of Φ.

Even a zero element of Φ in an off-diagonal position, say, (j,k), would signify that x_k is a minimizer of both $f_j(x)$ and $f_k(x)$, which could make x_k or its nearby points very desirable choices.

3.1.2 Quasi-normal instead of normal direction

The idea of a family of normals intersecting the boundary is valid even if we do not have the exact normal direction to the $CHIM$ simplex, but some *quasi-normal* direction $\hat{n}$ which points towards the origin. 'Shooting' a family of quasi-normal rays towards the boundary also gets us our desired boundary points. In practice we choose our quasi-normal direction to be an equally-weighted linear combination of the columns of Φ, multiplied by -1 to ensure that it points towards the origin. Explicitly,

$$\hat{n} = -\Phi e,$$

where e is the column vector of all ones.

The quasi-normal component defined as above has the property that the NBI point found for a certain w is completely independent of the scales of the objective functions. In other words, if NBI_w is re-solved with the objective functions rescaled by arbitrary factors, the NBI point found remains

unchanged. This fact will be proved later.

Given that Φ has nonnegative components as discussed in the previous subsection, it is clear that all components of Φe are nonnegative.

Even though a quasi-normal direction will be used in our computations, we prefer to retain the name 'NBI', rather than change it to something like 'QNBI'. The authors hope that this misnomer would not be considered too harshly.

3.1.3 Further insight: NBI and goal programming

Since t is being maximized in the NBI subproblem and $\Phi w + t\hat{n} = F(x)$, $x \in C$, this maximization subproblem attempts to find a feasible point x as far from a 'target' point Φw as possible, with $\hat{n} \geq 0$ (componentwise) guaranteeing nonincrease in the components of $F(x)$ relative to the components of Φw.

This is similar to goal programming. If we take the Pareto set to be convex in the objective space, 'equality goal programming'[3] can be thought of as NBI where the direction $\hat{n}$ is one of the canonical basis vectors e_i (i.e. with 1 in the ith position and 0 in the rest). To be precise, the subproblem NBI_w with $\hat{n} = e_i$ has the same solution as the following goal programming problem:

$$\min_x f_i(x)$$

$$s.t. \qquad f_j(x) = (\Phi w)(j), \quad j = 1, \ldots, n, \;\; j \neq i$$

$$x \in C,$$

where $(\Phi w)(j)$ denotes the jth component of the vector Φw.

Though posing the goals as equalities is untraditional, this kind of subproblem above for obtaining a Pareto optimal point is discussed in Lin[3] and [9].

---

[3]Preferring to goal programming where the goal constraints are equalities instead of inequalities.

7

3.1.4 Efficiently solving the subproblems

The following simple observation plays a key role in lowering the computational expense involved in solving the NBI subproblems:

Consider weight vectors w and $\bar{w}$ such that w is 'close to' $\bar{w}$, i.e., $\|w - \bar{w}\|$ is 'small' in some norm. Then, it is reasonable to expect that the solution (x^*, t^*) of NBI_w and the solution $(\bar{x}^*, \bar{t}^*)$ of $NBI_{\bar{w}}$ are 'close to each other'. Assume that we have solved $NBI_{\bar{w}}$ first and already have the point $(\bar{x}^*, \bar{t}^*)$. Then with $(\bar{x}^*, \bar{t}^*)$ as the starting point for solving NBI_w, the NBI subproblem solver can be expected to converge in a few iterations at a fast local convergence rate[4]. It is this aspect of our algorithm that gives it the flavor of a continuation-type method.

Since we already have the individual minima of the functions, i.e., the vertices of the $CHIM$ simplex, we start at x_1^* and solve a 'nearby subproblem', and then a subproblem close to the one just solved, and so on.

Let us illustrate the above strategy for a biobjective problem. The weights w for only two objectives can be expressed as $[\beta, 1 - \beta]$, $\beta \in [0, 1]$. We can take β to assume the values:

$$[0, \delta, 2\delta, \ldots, k\delta]$$

where $\delta < 1$ is the (uniform) spacing between two consecutive w_1 values and $k = I[\frac{1}{\delta}]$, i.e. the greatest integer $\leq \frac{1}{\delta}$. Then the set of 'uniformly distributed' weights is given by $[\beta, 1 - \beta]$, where β ranges over the values as above.

Now, assuming $\delta << 1$ (say $\delta = 0.05$), the minimizer of $f_2(x)$, i.e., x_2^*, is expected to be a small perturbation of the solution to the NBI subproblem with $w = [\delta, 1 - \delta]$. Thus the NBI subproblem with this w is solved starting from x_2^*, and its solution is used as the starting point for solving the NBI subproblem with $w = [2\delta, 1 - 2\delta]$, and so on, until the last weight is reached.

Of course, 'ordering the subproblems' may not be so obvious for problems with more that two objective functions, but can still be achieved, as described in the next section.

---

[4]Q-quadratic if exact second derivatives are used, superlinear if a secant approximation like BFGS is used.

4 Generating w and ordering the subproblems for more than two objectives

In this section, we shall describe a (data) structure which simultaneously enables the generation of weights w and ordering the subproblems in a manner amenable not only to efficient solution but also to parallelization.

4.1 Generating w

Let us assume that for an n-objective problem, $\delta_j > 0$ is the uniform spacing between two consecutive w_j values (i.e., the 'stepsize' on the jth component of w) for $j = 1, \ldots, n - 1$. For simplicity, let us also assume that $\frac{1}{\delta_1}$ is an integer.

The possible values that can be assumed by w_1 are

$$[0, \delta_1, 2\delta_1, \ldots, 1].$$

Define $m_1 = \frac{w_1}{\delta_1}$. Then the possible values of w_2 corresponding to $w_1 = m_1\delta_1$ (all the w_i's must add up to 1) are

$$[0, \delta_2, 2\delta_2, \ldots, k_2\delta_2]$$

where $k_2 = I\left[\frac{1-w_1}{\delta_2}\right] = I\left[\frac{1-m_1\delta_1}{\delta_2}\right]$.

Now define $m_2 = \frac{w_2}{\delta_2}$. Then the possible values of w_3 corresponding to $w_1 = m_1\delta_1$ and $w_2 = m_2\delta_2$ are

$$[0, \delta_3, 2\delta_3, \ldots, k_3\delta_3]$$

where $k_2 = I\left[\frac{1-w_1-w_2}{\delta_3}\right] = I\left[\frac{1-m_1\delta_1-m_2\delta_2}{\delta_3}\right]$.

Thus, corresponding to $w_i = m_i\delta_i$, $i = 1, \ldots, j - 1$, the possible values of w_j for $j = 2, \ldots, n - 1$ are

$$[0, \delta_j, 2\delta_j, \ldots, k_j\delta_j],$$

where

$$k_j = I\left[\frac{1 - \sum_{i=1}^{j-1} m_i\delta_i}{\delta_j}\right]$$

Finally the last component of w is defined as

$$w_n = 1 - \sum_{i=1}^{n-1} w_i.$$

Clearly, the entire data structure above can be thought of as a tree where the number of children varies with the node and generation. However, a tree structure is clearly unnecessary for implementation; all that requires storage are the numbers δ_j. However the tree is useful as a conceptual aid.

Of the subproblems generated by the weights in the above tree, n (with $w = e_i$) are already solved while finding F^*. Also note that since $\frac{\delta_i}{\delta_j}$ is not necessarily an integer $\forall i < j$, the spacings between 'the last two' values of w_n may not be uniform.

Special case: Equal stepsizes on all w_i

$$\text{Let } \delta_i = \delta, \ i = 1, \ldots, n - 1$$

Also assume that $\frac{1}{\delta} = p$ is an integer.

As before, the possible values of w_1 are

$$[0, \delta, 2\delta, \ldots, 1]$$

Then the possible values of w_j corresponding to $w_i = m_i \delta_i$, $i = 1, \ldots, j - 1$ for $j = 2, \ldots, n - 1$ are

$$[0, \delta, 2\delta, \ldots, (p - \sum_{i=1}^{j-1} m_i)\delta]$$

As before, $w_n = 1 - \sum_{i=1}^{n-1} w_i$, and now all the w_n values are uniformly spaced.

4.2   Ordering the subproblems

Each path from the root of the tree (the topmost node) to a leaf (a member in the bottommost generation) represents a unique weight w. It should also be observed that the w vectors are already ordered on the basis of 'nearness' as one traverses the tree breadthwise. Thus a strategy for picking the order

of the subproblems could be to start with the leftmost one (which has $w = e_n$ and is already solved) and solve the next one in the w_{n-1} generation (which is $w_{n-1} = \delta_{n-1}, w_n = 1 - \delta_{n-1}$), then the next one in the w_{n-1} generation ($w_{n-1} = 2\delta_{n-1}, w_n = 1 - 2\delta_{n-1}$), and so on until all the subproblems for $w_i = 0, i = 1, \ldots, n-2$ have been solved. Then we move to the next node in the w_{n-2} generation (i.e., with $w_i = 0, i = 1, \ldots, n-3$, $w_{n-2} = \delta_{n-2}$) and visit all the children of this node, with the starting points of the NBI subproblems chosen as the corresponding NBI subproblem solutions at the previous node.

This is where the scope for parallelization comes in. The solution of the first subproblem at the second node in the w_{n-2} generation didn't have to wait until all the subproblems in the first node were solved. The first subproblem in the second node of the w_{n-2} generation with $w_{n-2} = \delta_{n-2}$, $w_{n-1} = \delta_{n-1}$, $w_n = 1 - \delta_{n-2} - \delta_{n-1}$ could be solved immediately after solving the first subproblem in the first node with $w_{n-2} = 0$, $w_{n-1} = \delta_{n-1}$, $w_n = 1 - \delta_{n-1}$. Thus the first subproblem in the second node can be solved in parallel with the second subproblem in the first node, ..., and the kth subproblem in the second node can be solved in parallel with the $(k+1)^{th}$ subproblem of the first node. Further, the kth subproblem in the third node can be solved in parallel with the $(k+1)^{th}$ subproblem of the second node, with the solution of the kth subproblem of the second node as the starting point, and so on. This entire process of efficient parallelization is one of the topics of our future research.

5   Relationship between the NBI subproblem and minimizing a linear combination of the objectives

In this section we illustrate how the NBI subproblem is related to the popular method of minimizing a convex combination of the objectives. For ease of notation, we shall assume that the problem only has equality constraints, which can be assumed without loss of generality[5]. Let $\alpha \in (\Re_+ \cup \{0\})^n$, $\sum_1^n \alpha_i = 1$, denote a positive, convex weighting of the objectives. The weighted linear combination problem for obtaining a Pareto optimal point

---

[5] $h(x)$ can be thought of as the equality constraints augmented by the active set of inequality constraints and bounds

is then written as

$$\min_x \alpha^T F(x)$$

$$s.t. \quad h(x) = 0. \tag{1}$$

The solution of a problem like above will often be referred to as an *LC point*, and the problem denoted by LC_α. The 'first part' of the KKT conditions for optimality[6] of (x^*, λ^*) for problem (1) states that the gradient of the Lagrangian with respect to x should vanish at (x^*, λ^*), i.e.,

$$\nabla_x F(x^*)\alpha + \nabla_x h(x^*)\lambda^* = 0 \tag{2}$$

Similarly, if w denotes the vector of weights in NBI_w (which has a very different meaning from the weights α_i in the linear combinations subproblem), the NBI subproblem can be written as

$$\min_{x,t} -t$$

$$s.t. \quad F(x) - \Phi w - t\hat{n} = 0$$

$$h(x) = 0.$$

Then the first part of the KKT conditions states that the gradient of the Lagrangian with respect to (x, t) should vanish at $(x^*, t^*, \lambda^{(1)*}, \lambda^{(2)*})$, i.e.

$$\nabla_x F(x^*)\lambda^{(1)*} + \nabla_x h(x^*)\lambda^{(2)*} = 0 \tag{3}$$

$$-1 + \hat{n}^T \lambda^{(1)*} = 0,$$

where $\lambda^{(1)} \in \Re^n$ represents the vector of multipliers corresponding to the constraints $\Phi w + t\hat{n} - F(x) = 0$, and $\lambda^{(2)} \in \Re^{ne}$ denotes the multipliers of the equality constraints $h(x) = 0$.

Claim:
Suppose $(x^*, t^*, \lambda^{(1)*}, \lambda^{(2)*})$ is the solution of NBI_w. Now define the components of the vector α as

$$\alpha_i = \frac{\lambda_i^{(1)*}}{\sum_1^n \lambda_i^{(1)*}}.$$

---

[6]Karush-Kuhn-Tucker conditions, or alternately the first order necessary conditions for optimality.

Then, problem (1) with the above convex weighting vector α has the solution

$$[x^*, \lambda^* = \frac{1}{\sum_1^n \lambda_i^{(1)*}} \lambda^{(2)*}].$$

Proof:
Dividing both sides of (3) by the scalar $\sum_1^n \lambda_i^{(1)*}$ and observing that $h(x^*) = 0$, the equivalence between (2) and (3) becomes obvious.

However, quite clearly, if for some i, the sign of $\lambda_i^{(1)*}$ is opposite to that of $\sum_1^n \lambda_i^{(1)*}$, then the vector α has a negative component and does not qualify as a weight for problem (1). In such a case, either the Pareto optimality of the NBI point $(x^*, t^*, \lambda^{(1)*}, \lambda^{(2)*})$ is questionable, or the Pareto point lies in a nonconvex part of the Pareto set[7].
Also observe the tacit assumption that $\sum_1^n \lambda_i^{(1)*} \neq 0$.

Just as the analysis above suggests a method for obtaining α for problem LC_α given the corresponding solution of NBI_w, one can also obtain the NBI point corresponding to a given solution of problem LC_α with very little effort.

Suppose (x^*, λ^*) solves problem LC_α. Let $(\bar{w}, t^*)$ be the solution of the $(n+1) \times (n+1)$ linear system

$$\Phi w + t\hat{n} = F(x^*)$$

$$\sum_{i=1}^n w_i = 1.$$

Then (x^*, λ^*) corresponds to the solution of NBI_w with $w = \bar{w}$, i.e., the solution of $NBI_{\bar{w}}$ is

$$(x^*, t^*, \lambda^{(1)*} = \frac{\alpha}{\alpha^T \hat{n}}, \lambda^{(2)*} = \frac{\lambda^*}{\alpha^T \hat{n}}).$$

Proof:

---

[7]Pareto points in nonvonvex parts of the Pareto set cannot be obtained by minimizing a linear combination of the objectives, a proof of which will appear in a future article

Dividing (2) on both sides by $\alpha^T \hat{n}$ (assumed nonzero[8]) and observing that $\lambda^{(1)*}$ defined above satisfies $\hat{n}^T \lambda^{(1)*} = 1$, it can be seen that the first part of the KKT conditions for $NBI_{\bar{w}}$ holds. Further observing that, $h(x^*) = 0$ and $\Phi w + t\hat{n} = F(x^*)$, the required equivalence between LC_α and $NBI_{\bar{w}}$ follows.

6 Proof of independence with respect to function scales using the quasi-normal

In this section we shall prove that the NBI point found using the quasi-normal $\hat{n}$ and a particular w is independent of how the individual functions are scaled.

Let the objective functions be scaled by positive scalars s_i as

$$f_i(x) \leftarrow s_i f_i(x), \quad i = 1, \ldots, n.$$

In other words, if s is the vector with components s_i and $S = diag(s)$, then

$$F(x) \leftarrow SF(x).$$

Consequently

$$\nabla_x F(x) \leftarrow \nabla_x F(x)S$$

$$\Phi = S\Phi$$

The quasi-normal direction $\hat{n} = -\Phi e$ after scaling becomes $= -S\Phi e$.

Claim:

If $(x^*, t^*, \lambda^{(1)*}, \lambda^{(2)*})$ solves the unscaled NBI_w (i.e. with $S = I_n$), then $(x^*, t^*, S^{-1}\lambda^{(1)*}, \lambda^{(2)*})$ solves[9] NBI_w with the functions scaled as above.

Proof: Since $(x^*, t^*, \lambda^{(1)*}, \lambda^{(2)*})$ solves the unscaled NBI_w (still with only equality constraints as in the previous section),

$$\nabla_x F(x^*)\lambda^{(1)*} + \nabla_x h(x^*)\lambda^{(2)*} = 0$$

---

[8]Since α has nonnegative components (not all zero) and $\hat{n}$ has negative components, the assumption holds.

[9]Here 'solves' means 'finds a stationary point of the nonlinear programming problem'.

$$\hat{n}^T \lambda^{(1)*} = 1$$

$$\Phi w + t^* \hat{n} = F(x^*)$$

$$h(x^*) = 0.$$

The first equation can be rewritten to state that the following holds:

$$(\nabla_x F(x^*)S)(S^{-1}\lambda^{(1)*}) + \nabla_x h(x^*)\lambda^{(2)*} = 0. \tag{4}$$

The second equation implies

$$e^T \Phi^T \lambda^{(1)*} = 1$$

$$\equiv e^T \Phi^T S S^{-1} \lambda^{(1)*} = 1.$$

Since $S = S^T$, the above is the same as

$$(e^T (S\Phi)^T)(S^{-1}\lambda^{(1)*}) = 1. \tag{5}$$

The third equation can be rewritten as

$$\Phi w + t^* \Phi e = F(x^*)$$

$$\equiv S\Phi w + t^* S\Phi e = SF(x^*). \tag{6}$$

Clearly, equations (4),(5) & (6) imply that $(x^*, t^*, S^{-1}\lambda^{(1)*}, \lambda^{(2)*})$ solves NBI_w with the functions scaled by S.
(QED)

Tha above result does not depend on e being the vector of all ones and consequently holds if $\hat{n}$ is scaled by a factor, say, a normalization constant.

The above result suggests that no matter how disparately the different functions might be scaled, NBI with the quasi-normal finds a set of points as if the functions were all scaled to the same order of magnitude.

7 Advantages of using NBI

- **Finds a uniform spread of Pareto points:** Consider any method which parametrically combines all the objective functions into a single objective and finds efficient points by minimizing the single objective for various values of the parameters. Then, in general, the mapping

15

from the set of parameters to the set of Pareto optimal points is *not* one-to-one. Thus it might so happen that minimizations over several different parameters produces the very same point each time, resulting in fruitless computational expense– this is never the case with NBI. Moreover, in the absence of convexity, "Pareto-optimal solutions obtained by this method are often found to be so few, or the corresponding indexes so extreme, that there seems to be no middle 'ground' for any compromise, although such 'ground' may actually exist" – Lin [9]. For examples, refer to Lin [9], Katopis and Lin [10], Lin [11].

The interrelationship between the linear combinations subproblem and the NBI subproblem provides more insight into why the linear combinations technique fail to give a uniformly distributed set of Pareto optima. By fixing the weights α in subproblem LC, we are in effect fixing the multipliers of the corresponding NBI subproblem, thus partly restricting the solution of the resultant subproblem. Even if the Pareto optima are uniformly distributed in the Pareto set, there is no reason why the corresponding multipliers have to be uniformly distributed. However, the weights in the linear combinations approach are often very desirable because they give an idea of the relative importance of the objectives. Thus obtaining the NBI points, which are uniformly distributed, and then finding the corresponding weights α for the NBI points can be very useful.

- **Advantages over homotopy techniques:** NBI improves over homotopy/continuation techniques for tracing the curve of Pareto optimal solutions, like the one discussed in Rakowska, Haftka & Watson [4], in the following respects:

  - *It is applicable for more than two objectives* For a multiobjective problem with more than two objectives the homotopy parameter is not a scalar and the associated differential equations turn out to be a system of nonlinear partial differential equations with not readily available boundary conditions, rather than an ordinary initial value problem, as in the case of two objectives. Thus extending homotopy techniques to handle $n > 2$ is very difficult. On the other hand, NBI can be extended to handle more than two objectives quite easily.

  - *It does not require exact Hessian.* Even for a biobjective problem, solving the homotopy boundary value problem requires exact sec-

16

ond derivative information (i.e., the Hessian of the Lagrangian), whereas the NBI subproblem solver requires only a secant approximation of the Hessian like BFGS.

– *It can bypass tracking active sets.* For problems with inequality constraints or explicit bounds on variables , homotopy techniques need to keep track of the changes in active sets of the inequality constraints or bounds meticulously in course of the Initial Value Problem integration, which can present difficulties if the number of inequalities or bounds is large. On the other hand an interior point NLP solver used as the NBI subproblem solver would handle this situation quite efficiently, and would not have a problem with frequent changes in the active set.

• NBI improves on other traditional methods like goal programming in the sense that it never requires any prior knowledge of 'feasible goals'. It improves on multilevel optimization techniques from the tradeoff standpoint, since multilevel techniques usually can only improve only a few of the 'most important' objectives, leaving no compromise for the rest.

8   A note on local versus global

It is worth observing here that unless the individual minima of the objectives obtained at the outset are guaranteed to be global minima there is no guarantee that NBI produces solutions that are globally Pareto optimal. In fact, as pointed out earlier, there is no guarantee that every solution produced by NBI is even locally Pareto optimal. All we can conjecture is that if the individual minima of the functions happen to be global minima and if we start NBI from every point on $CHIM + \cup CHIM$, the set of points thus obtained would contain all the globally Pareto optimal points, provided the boundary of $\mathcal{F}$ is not 'folded'. However, even when 'folded', the point obtained could be locally Pareto optimal (see fig.2).

Not being able to find globally Pareto optimal points is a drawback inherent in every method that finds a large number of efficient points of MOP. In homotopy methods, it would involve finding the global minimum of one of the two objectives in the very beginning. In methods which find efficient points by minimizing a single objective, only a global minimum of the scalarized objective would correspond to a globally efficient point. Even though
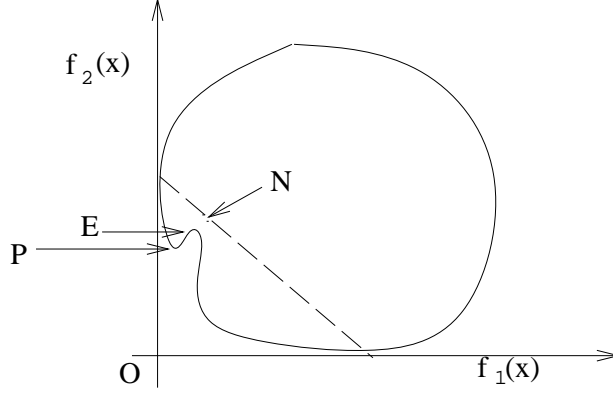
Figure 3: The normal from N intersects the boundary at E, but values of the objectives at P are each less than the corresponding values at E, hence E is not Pareto optimal.

a local minimum would still correspond to a locally efficient point, there is no guarantee that minimizing a single objective produces a local minimum since most single objective optimization algorithms only converge to a KKT point of the problem, i.e. one which only satisfies necessary conditions for being a minimum and could thus well be a saddle-point (and not even a local minimum!).

Given the shortcomings of global optimization applied to nonconvex problems, we choose to remain satisfied with the Pareto optimal points obtained by NBI, in spite of the fact that they may not be globally efficient.

9   A Numerical Example

Below is a brief account of employing NBI techniques on a small biobjective problem, stated below:

$$\min_x \left[\begin{array}{c} f_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \\ f_2(x) = 3\,x_1 + 2\,x_2 - \frac{x_3}{3} + 0.01\,(x_4 - x_5)^3 \end{array} \right]$$

$$s.t. \quad x_1 + 2\,x_2 - x_3 - 0.5\,x_4 + x_5 = 2$$

$$4\,x_1 - 2\,x_2 + 0.8\,x_3 + 0.6\,x_4 + 0.5\,x_5^2 = 0$$

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 \leq 10.$$

NBI using the actual normal to the $CHIM$ simplex (a line segment in this case) was run three times on this problem for 21 different weight vectors w: first on the original problem, then on the problem with f_1 scaled by a factor of 5 (to increase the disparity between the scales of the objective functions) and then with f_1 scaled by a factor of 10. The results in the following table shows that NBI succesfully produces a uniformly distributed set of Pareto optimal points even if the objective functions are scaled disparately. (Note that the tabulated Pareto optimal function values have all been converted back to their original scales.)

Weights (w_1, w_2)	Objective values (original scale)	Objective values (f_1 scaled by 5)	Objective values (f_1 scaled by 10)
0.00 , 1.00	10.0000 , -4.0111	10.0000, -4.0111	10.0000, -4.0111
0.05 , 0.95	9.4717 , -3.7902	9.5249, -3.8126	9.5270, -3.8135
0.10 , 0.90	8.9453 , -3.5665	9.0499, -3.6113	9.0541, -3.6131
0.15 , 0.85	8.4208 , -3.3398	8.5750, -3.4069	8.5812, -3.4095
0.20 , 0.80	7.8985 , -3.1097	8.1002, -3.1991	8.1083, -3.2027
0.25 , 0.75	7.3785 , -2.8759	7.6255, -2.9876	7.6354, -2.9921
0.30 , 0.70	6.8612 , -2.6381	7.1508, -2.7720	7.1626, -2.7773
0.35 , 0.65	6.3469 , -2.3958	6.6763, -2.5517	6.6897, -2.5580
0.40 , 0.60	5.8359 , -2.1483	6.2020, -2.3263	6.2170, -2.3335
0.45 , 0.55	5.3286 , -1.8951	5.7277, -2.0950	5.7442, -2.1032
0.50 , 0.50	4.8256 , -1.6353	5.2537, -1.8570	5.2715, -1.8661
0.55 , 0.45	4.3275 , -1.3679	4.7799, -1.6112	4.7989, -1.6213
0.60 , 0.40	3.8353 , -1.0916	4.3063, -1.3562	4.3263, -1.3672
0.65 , 0.35	3.3499 , -0.8046	3.8329, -1.0903	3.8538, -1.1022
0.70 , 0.30	2.8730 , -0.5047	3.3600, -0.8107	3.3813, -0.8237
0.75 , 0.25	2.4067 , -0.1885	2.8875, -0.5141	2.9090, -0.5281
0.80 , 0.20	1.9542 , 0.1490	2.4155, -0.1947	2.4368, -0.2097
0.85 , 0.15	1.5209 , 0.5159	1.9444, 0.1567	1.9649, 0.1406
0.90 , 0.10	1.1164 , 0.9272	1.4747, 0.5586	1.4932, 0.5413
0.95 , 0.05	0.7635 , 1.4178	1.0074, 1.0583	1.0222, 1.0398
1.00 , 0.00	0.5551 , 2.1306	0.5551, 2.1306	0.5551, 2.1306

The plots of Pareto optimal objective vectors as tabulated above for the original and scaled problems as shown in Fig.4 and Fig.5, reveal very slight difference: with the first objective scaled, one point on the $F(x_1^*)$ end

moves a little further away. However, using the quasi-normal $\hat{n}$, even this slight nonuniformity of distribution of Pareto points is eliminated (see Fig. 6). The Pareto points obtained using the quasi-normal, independent of the scale on f_1, and are tabulated below:

Weights	Objective values
0.00, 1.00	10.0000 , -4.0111
0.05, 0.95	9.4254 , -3.7706
0.10, 0.90	8.8546 , -3.5276
0.15, 0.85	8.2882 , -3.2818
0.20, 0.80	7.7264 , -3.0329
0.25, 0.75	7.1698 , -2.7807
0.30, 0.70	6.6189 , -2.5247
0.35, 0.65	6.0743 , -2.2647
0.40, 0.60	5.5368 , -2.0000
0.45, 0.55	5.0072 , -1.7302
0.50, 0.50	4.4866 , -1.4546
0.55, 0.45	3.9764 , -1.1722
0.60, 0.40	3.4781 , -0.8820
0.65, 0.35	2.9939 , -0.5827
0.70, 0.30	2.5266 , -0.2724
0.75, 0.25	2.0801 , 0.0514
0.80, 0.20	1.6597 , 0.3922
0.85, 0.15	1.2740 , 0.7556
0.90, 0.10	0.9370 , 1.1506
0.95, 0.05	0.6754 , 1.5947
1.00, 0.00	0.5551 , 2.1306

The method of linear combinations was run thrice on the same problem, with the weight vectors α assuming the same 21 uniformly spread values as the w vector above[10].

When run on the original problem, the minimizer of $f_2(x)$ was found six times for six different α, and there was a considerable gap 'in the middle' of the Pareto set [see fig.(7)].

With f_1 scaled by 5, the point found six times earlier was found only twice[11],

---

[10]The efficient solution scheme, i.e., starting the solution of a subproblem from the optimal point of a 'nearby subproblem' was used here too.

[11]Heavily weighting the first objective made the minimizer move away from x_2^*.

but the Pareto optimal vectors obtained were concentrated at the $F(x_1^*)$ end and no 'middle ground for compromise' was captured [see fig.(8)].
With f_1 scaled by 10, the point repeated earlier was found only once, though the clustering at the $F(x_1^*)$ end increased [see fig.(9)].

The Pareto optimal vectors obtained using linear combinations are tabulated below:

Weights (α_1, α_2)	Objective values (original scale)	Objective values (f_1 scaled by 5)	Objective values (f_1 scaled by 10)
0.00 , 1.00	10.0000 , -4.0111	10.0000, -4.0111	10.0000, -4.0111
0.05 , 0.95	10.0000 , -4.0111	10.0000, -4.0111	4.8211, -1.6330
0.10 , 0.90	10.0000 , -4.0111	4.1857, -1.2896	1.1634, 0.8741
0.15 , 0.85	10.0000 , -4.0111	1.6131, 0.4330	0.7689, 1.4083
0.20 , 0.80	10.0000 , -4.0111	1.0180, 1.0451	0.6559, 1.6416
0.25 , 0.75	10.0000 , -4.0111	0.7975, 1.3592	0.6100, 1.7724
0.30 , 0.70	8.9403 , -3.5644	0.6953, 1.5506	0.5876, 1.8563
0.35 , 0.65	4.5379 , -1.4822	0.6412, 1.6796	0.5754, 1.9146
0.40 , 0.60	2.7307 , -0.4109	0.6100, 1.7725	0.5682, 1.9576
0.45 , 0.55	1.8319 , 0.2473	0.5909, 1.8425	0.5637, 1.9905
0.50 , 0.50	1.3357 , 0.6928	0.5788, 1.8973	0.5608, 2.0165
0.55 , 0.45	1.0425 , 1.0147	0.5707, 1.9413	0.5589, 2.0376
0.60 , 0.40	0.8615 , 1.2583	0.5654, 1.9773	0.5576, 2.0551
0.65 , 0.35	0.7463 , 1.4492	0.5618, 2.0075	0.5567, 2.0698
0.70 , 0.30	0.6719 , 1.6029	0.5593, 2.0331	0.5561, 2.0823
0.75 , 0.25	0.6236 , 1.7295	0.5576, 2.0551	0.5557, 2.0931
0.80 , 0.20	0.5926 , 1.8356	0.5565, 2.0741	0.5554, 2.1025
0.85 , 0.15	0.5734 , 1.9258	0.5558, 2.0909	0.5553, 2.1108
0.90 , 0.10	0.5622 , 2.0035	0.5554, 2.1057	0.5552, 2.1181
0.95 , 0.05	0.5567 , 2.0711	0.5551, 2.1188	0.5551, 2.1247
1.00 , 0.00	0.5551 , 2.1306	0.5551, 2.1306	0.5551, 2.1306

Clearly, the inability of the method of linear combinations in sufficiently capturing the 'middle ground' of the Pareto set renders it fairly useless as a means of studying the tradeoff between the conflicting objectives.

9.1 Function scaling implicit in NBI

Even though the NBI using the quasi-normal component is unaffected by the function scales, this property comes with a price. As the functions get more disparately scaled, the Pareto set gets more 'stretched', and consequently the NBI points get further apart from each other. Consequently, solving an NBI subproblem starting from the solution of the same nearby subproblem takes more iterations to converge. This was observed in the numerical example above and motivates the need to scale the functions properly to remove this disparity in scales.

Geometrically, it can be perceived that if the vertices of the $CHIM$ simplex are almost equidistant from the origin, i.e. the quantities

$$\|F(x_j^*) - F^*\|, \quad j = 1, \ldots, n$$

are almost equal, then the quasi normal direction $\hat{n}$ is almost normal to the $CHIM$ simplex. This would achieve the 'minimally stretched' Pareto set we want and could also be a good scaling for the problem in the sense that all the functions would be about the same order of magnitude, and thus reduce possible ill-conditioning.

For the biobjective problem, Φ is antidiagonal; thus a scaling that would achieve the above is obvious:

$$f_1 \leftarrow \frac{f_1}{f_1(x_2^*)}$$

$$f_2 \leftarrow \frac{f_2}{f_2(x_1^*)},$$

which gets each vertex of $CHIM$ to be unit distance from the origin.

However, the solution may not be so transparent for more than two objectives, and it may not be possible to get all the vertices exactly equidistant from the origin. So now we shall attempt to find function scalings $d_i > 0$ such that the functions scaled as

$$f_i \leftarrow \sqrt{d_i} f_i$$

will have the property that the *variance* among the scaled distances of the vertices from the origin, i.e.

$$\|\sqrt{D}(F(x_j^*) - F^*)\|^2, \quad j = 1, \ldots, n$$

will be minimized ($D = diag(d)$, d represents the vector with components d_i).

Let $v_j = \|\sqrt{D}(F(x_j^*) - F^*)\|^2$, i.e.,

$$v_j = \sum_{i=1}^{n} d_i \phi_{i,j}^2,$$

where $\phi_{i,j}$ is the ith row j^{th} column entry of the matrix Φ.

The mean square distance of the vertices is defined as

$$\bar{v} = \frac{1}{n} \sum_{j=1}^{n} v_j = \frac{1}{n} \sum_{i=1}^{n} d_i \left(\sum_{j=1}^{n} \phi_{i,j}^2 \right).$$

The variance quantity to be minimized is given by

$$V(d) = \sum_{j=1}^{n} (v_j - \bar{v})^2;$$

i.e.,

$$V(d) = \sum_{j=1}^{n} \left\{ \sum_{i=1}^{n} d_i \phi_{i,j}^2 - \sum_{i=1}^{n} d_i \left(\frac{1}{n} \sum_{j=1}^{n} \phi_{i,j}^2 \right) \right\}^2.$$

Let A be the matrix with components $a_{i,j}$ given by

$$a_{i,j} = \phi_{i,j}^2 - \frac{1}{n} \sum_{k=1}^{n} \phi_{i,k}^2.$$

Then

$$V(d) = \sum_{j=1}^{n} \left(\sum_{i=1}^{n} d_i a_{i,j} \right)^2;$$

i.e.,

$$V(d) = d^T A A^T d = \|A^T d\|^2.$$

This quadratic function is convex in d, and has an unconstrained minimizer at $d = 0$. Thus we shall demand a specific value of $\bar{v}$, which represents an average distance of the CHIM simplex from the origin[12] and is roughly

_____

[12]Using the mean distance instead of the mean square distance for this constraint would result in loss of convexity.

the same order of magnitude as a typical function value of any objective encountered in the computation. Say we want a typical objective value to be τ, which could be something like 10. Then we would enforce

$$\bar{v} = \frac{1}{n} \sum_{i=1}^{n} d_i (\sum_{j=1}^{n} \phi_{i,j}^2) = \tau$$

along with a small lower bound on d_i. Thus the optimization problem to be solved to obtain our 'optimal' scales is

$$\min_{d} V(d) = d^T A A^T d$$

$$s.t. \qquad \sum_{i=1}^{n} d_i (\sum_{j=1}^{n} \phi_{i,j}^2) = n\tau$$

$$d_i >= 10^{-8} \quad, i = 1, \ldots, n.$$

Thus we can see how the matrix Φ suggests an 'improved scaling' of the objective functions, which is a bonus in the NBI approach.

10 Conclusion

An algorithm was presented for finding Pareto optimal points of any smooth, constrained multiobjective problem with essentially any number of objectives. One question that is left open is how the user would select the final design point from the Pareto set generated by NBI (or any other algorithm which generates the Pareto set). For two or three objectives, the generated Pareto curve/surface can be visualized with standard 2-D or 3-D plots, which may be all the user needs to arrive at a final design point. However the visualization process may be complicated for more than three objectives, and how helpful it will be in guiding the user towards a better choice may depend on factors like the psychological aspects of the visualization. One procedure that could perhaps be useful is to have the user specify another 'cost' or 'utility' function, whose value could be reported at each of the Pareto optimal points generated by NBI, and the user could make his/her final choice based on this 'cost'. Also, if there are more than three objectives and if it is possible to set up a hierarchical order of preference in blocks of two or three (e.g. f_2, f_4, f_5 are more important than f_1, f_3), the Pareto points for the combined problem could be visualized for each of the blocks, starting at the most important, and the user could narrow down his/her

preferences down the blocks.

Further research is in progress regarding the above issue and also regarding the development of efficient nonlinear programming techniques for solving the NBI subproblems and parallelizing the entire algorithm.

11   Acknowledgements

References

[1] H. Eschenauer, J. Koski and A. Osyczka. *Multicriteria Design Optimization*. Berlin, Springer-Verlag, 1990.

[2] Roman B. Statnikov and Joseph B. Matusov. *Multicriteria Optimization and Engineering*. New York, Chapman & Hall, 1995.

[3] J. G. Lin. *Three Methods for Determining Pareto-Optimal Solutions of Multiple-Objective Problems*. Directions in Large-Scale Systems, pp. 117-138. Edited by Y. C. Ho and S. K. Mitter. New York, Plenum Press, 1975.

[4] J. Rakowska, R. T. Haftka and L. T. Watson. *Tracing the Efficient Curve for Multi-Objective Control-Structure Optimization*. Computing Systems in Engineering. Vol. 2, No. 6, pp. 461-471, 1991.

[5] J. R. Rao and P. Y. Papalambros. *A Non-linear Programming Continuation Strategy for One Parameter Design Optimization Problems*. Proceedings of ASME Design Automation Conference, Montreal, Quebec, Canada, Sept. 17-20, 1989, pp. 77-89.

[6] B. N. Lundberg and A. B. Poore. *Bifurcations and Sensitivity in Parametric Programming*. Proceedings of Third Air Force/NASA Symposium on Recent Advances in Multidisciplinary Analysis and Optimization, Sept. 24-26, 1990, San Francisco, CA, pp. 50-55.

[7] J. Koski. *Multicriteria Truss Optimization*. Multicriteria Optimization in Engineering and in the Sciences. Edited by W. Stadler. New York, Plenum Press, 1988.

[8] Y. Haimes, W. Hall and H. Freedman. *Multiobjective Optimization in Water Resources Systems*. Amsterdam, Elsevier Scientific Publishing Co, 1975.

[9] J. G. Lin. *Multiple-Objective Problems: Pareto-Optimal Solutions by Method of Proper Equality Constraints*. IEEE Transactions on Automatic Control, vol. AC-21, no.5, October 1976, pp. 641-650.

[10] G. A. Katopis and J. G. Lin. *Non-inferiority of Controls under Double Performance Objectives: Minimal Time and Minimal Energy*. Proceedings of 7th Hawaii Int. Conf. Syst. Sci., Honolulu, Hawaii, Jan 1974, pp. 129-131.

[11] J. G. Lin. *Circuit Design under Multiple Performance Objectives*. Proc. 1974 IEEE Int. Symp. Circuits & Systems, San Francisco, CA, pp. 549-552, Apr. 1974.
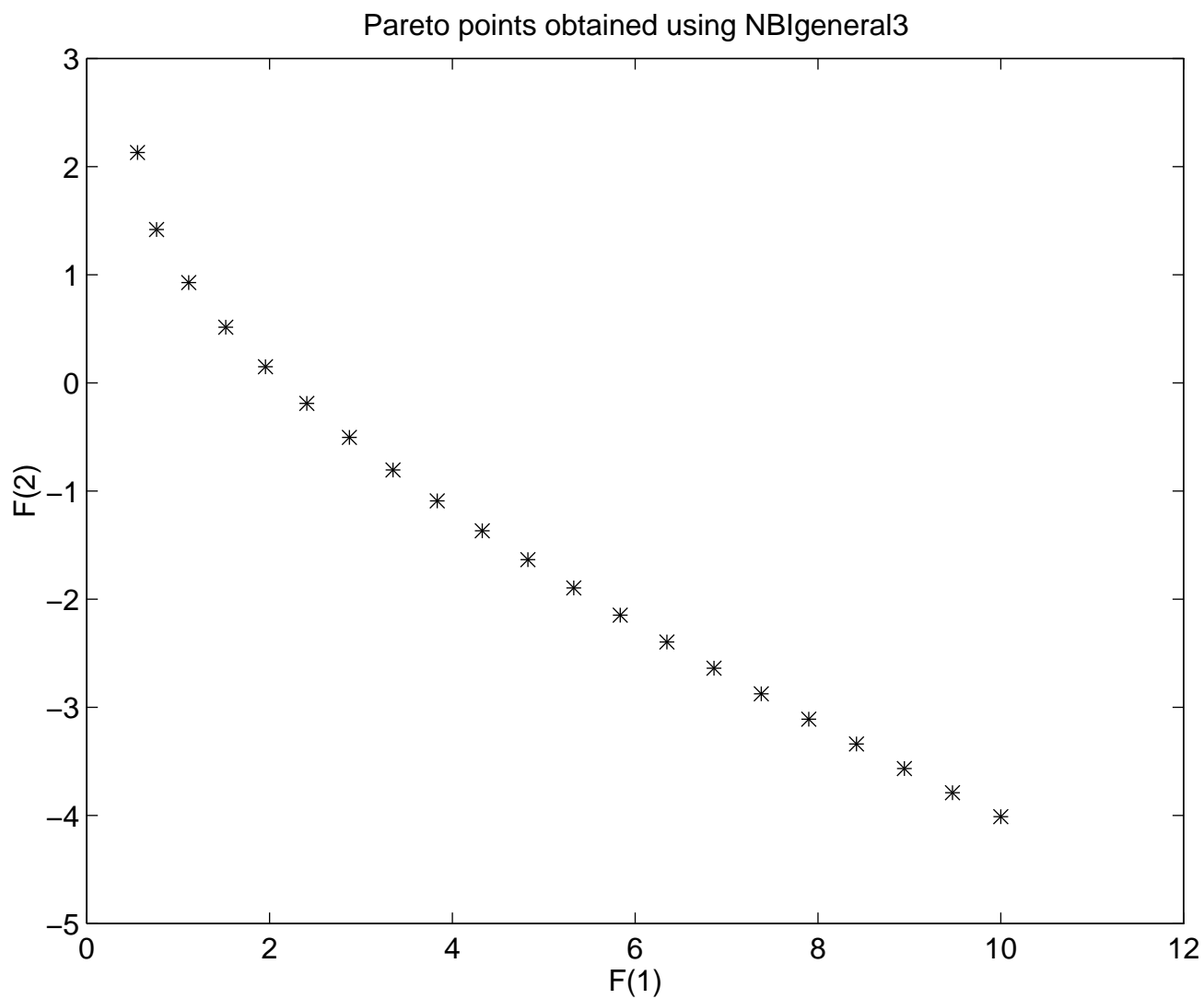
Figure 4: Pareto optimal vectors in the objective space using NBI with actual normal on the original problem
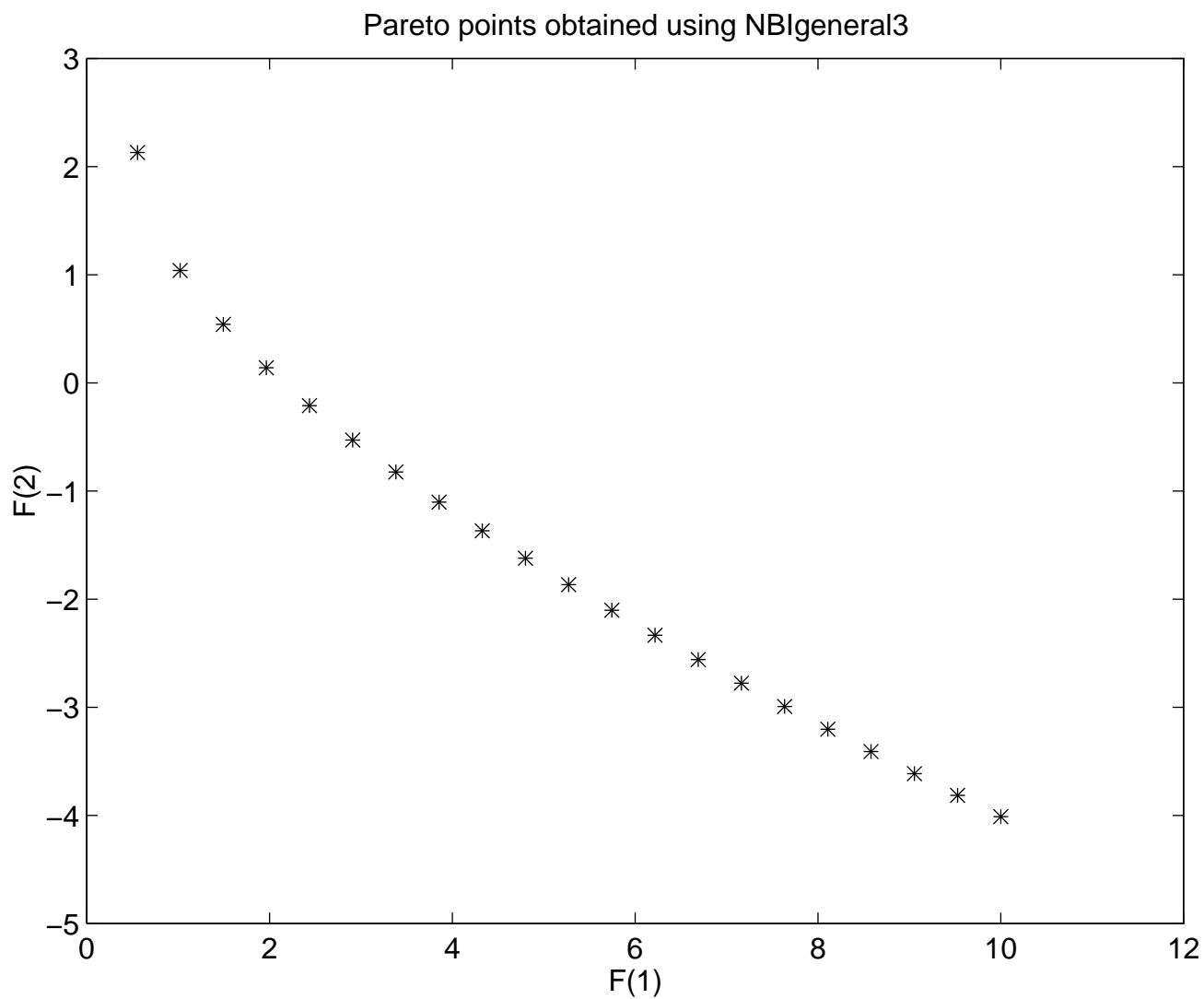
Figure 5: Pareto optimal vectors in the objective space using NBI with actual normal on the problem with f_1 scaled by 10
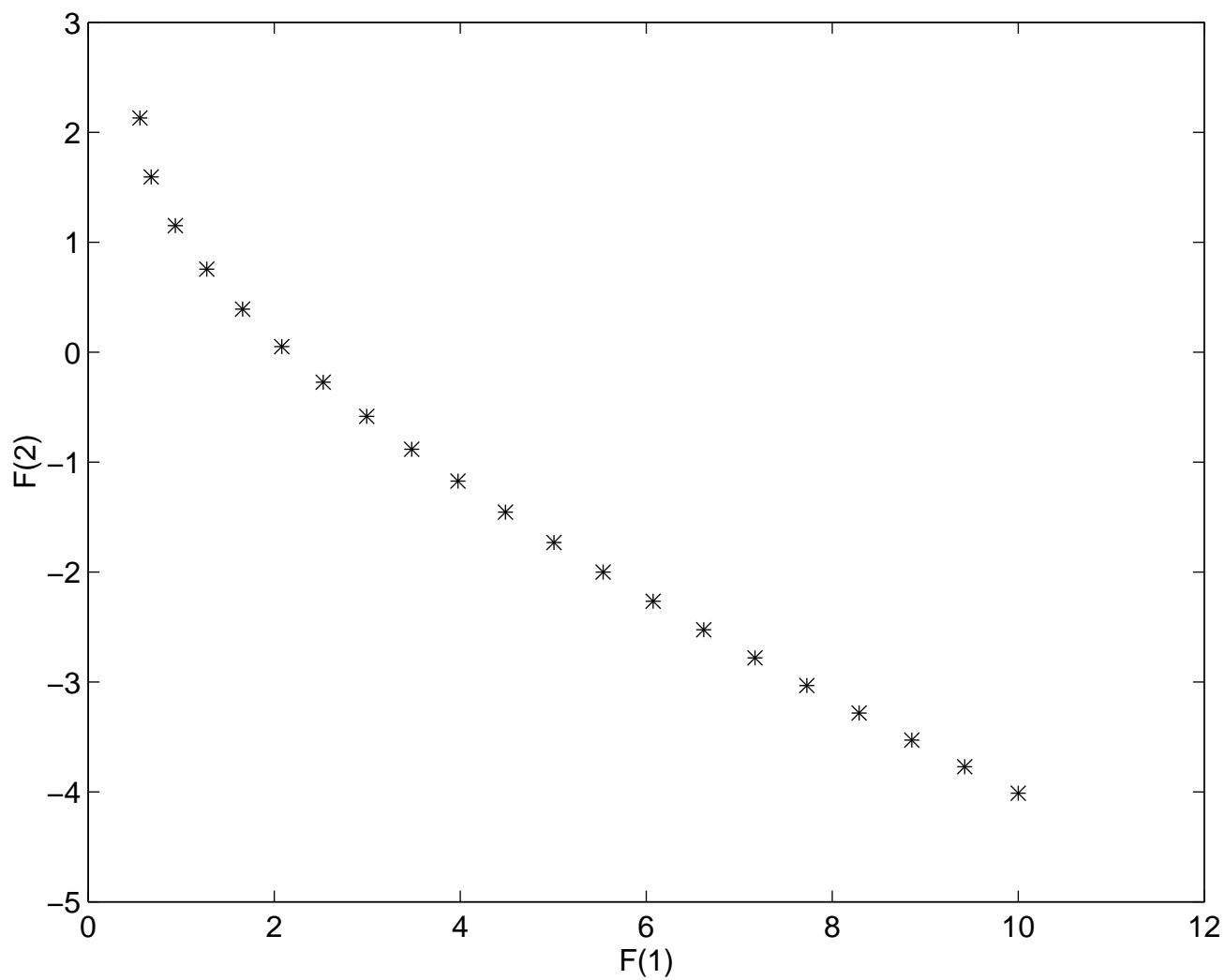
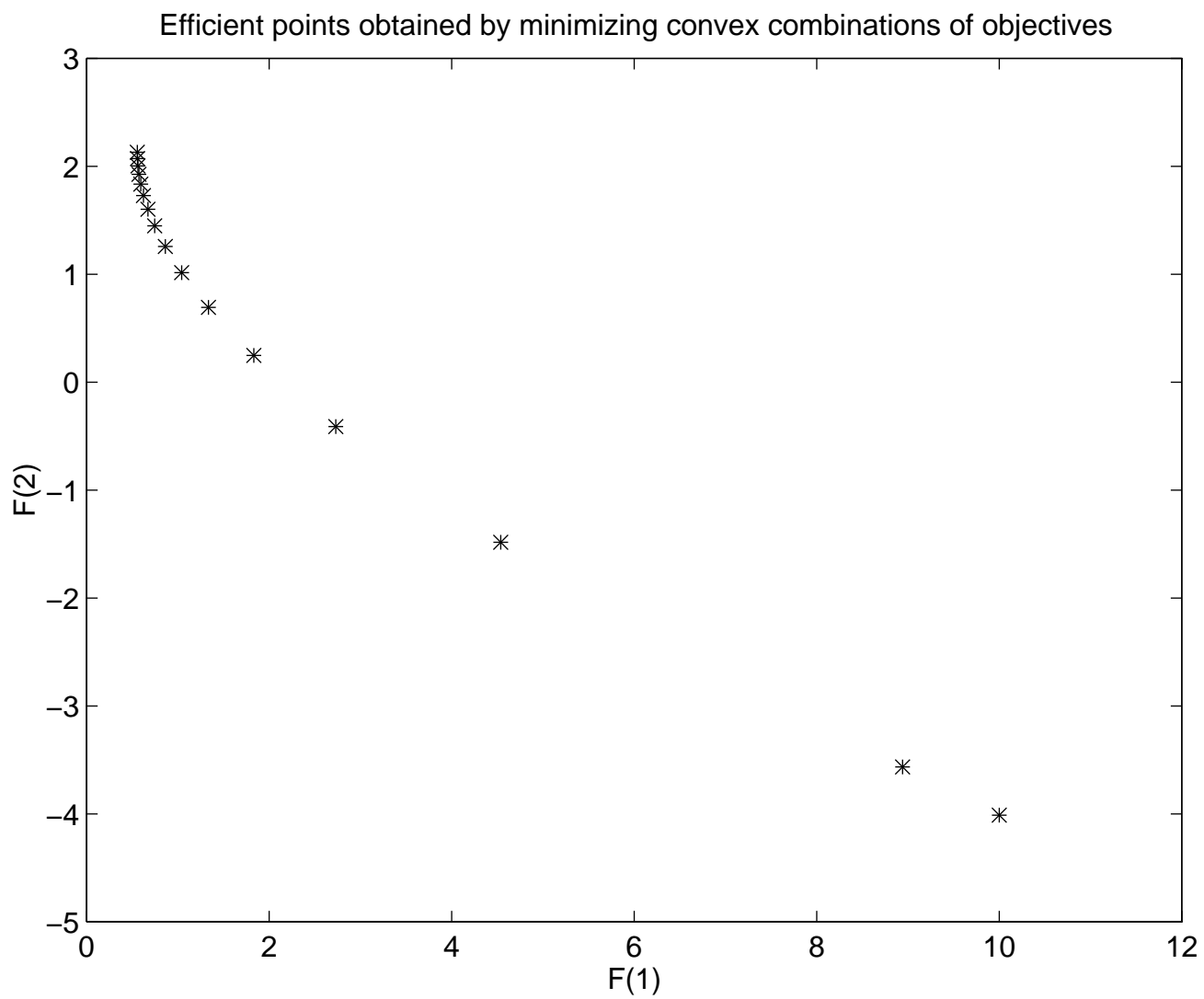Figure 6: Pareto optimal vectors in the objective space using NBI with quasi-normal on the problem with f_1 scaled by 10

Figure 7: Pareto optimal vectors in the objective space using the method of linear combinations on the original problem
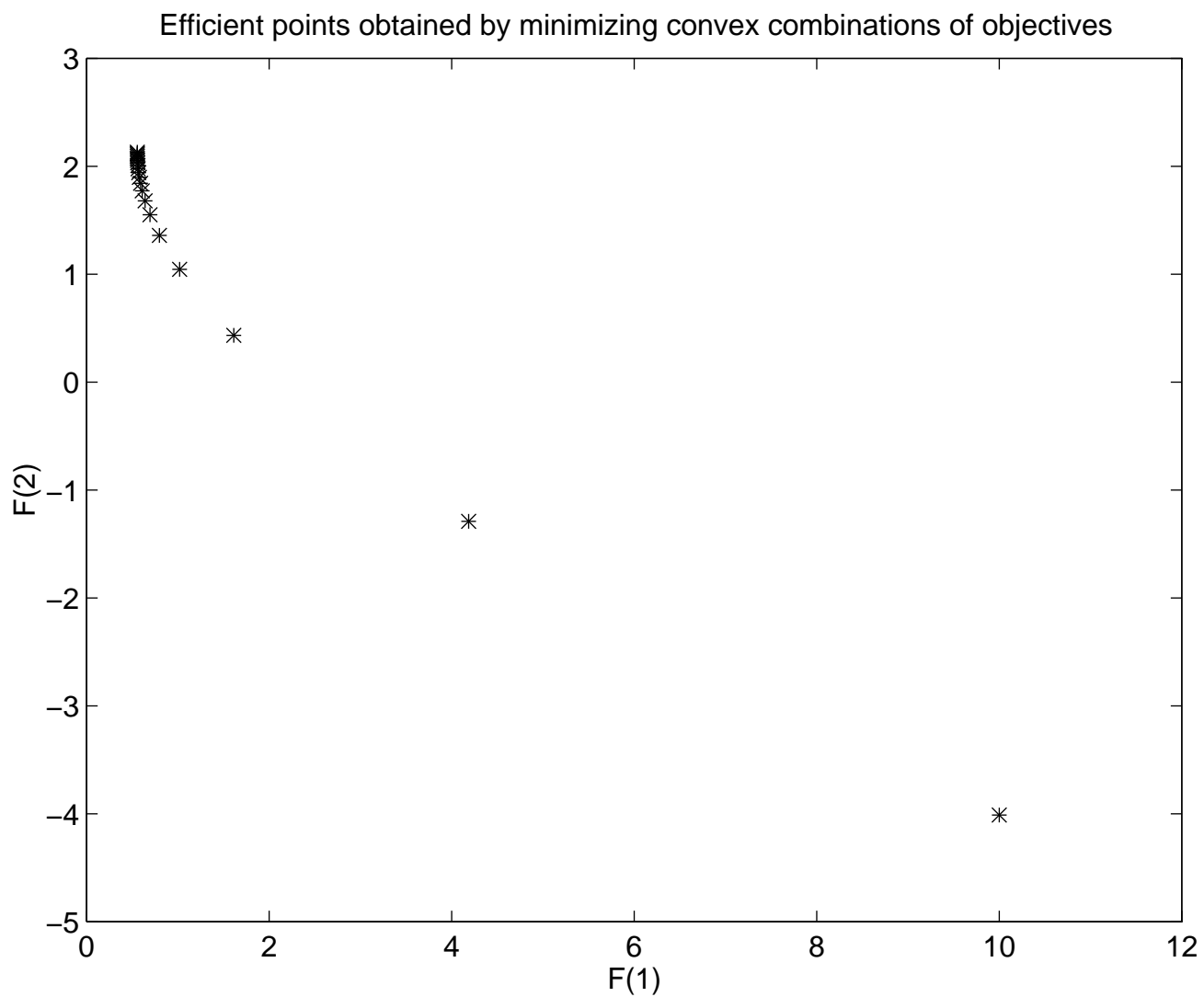
Figure 8: Pareto optimal vectors in the objective space using the method of linear combinations on the problem with f_1 scaled by 5
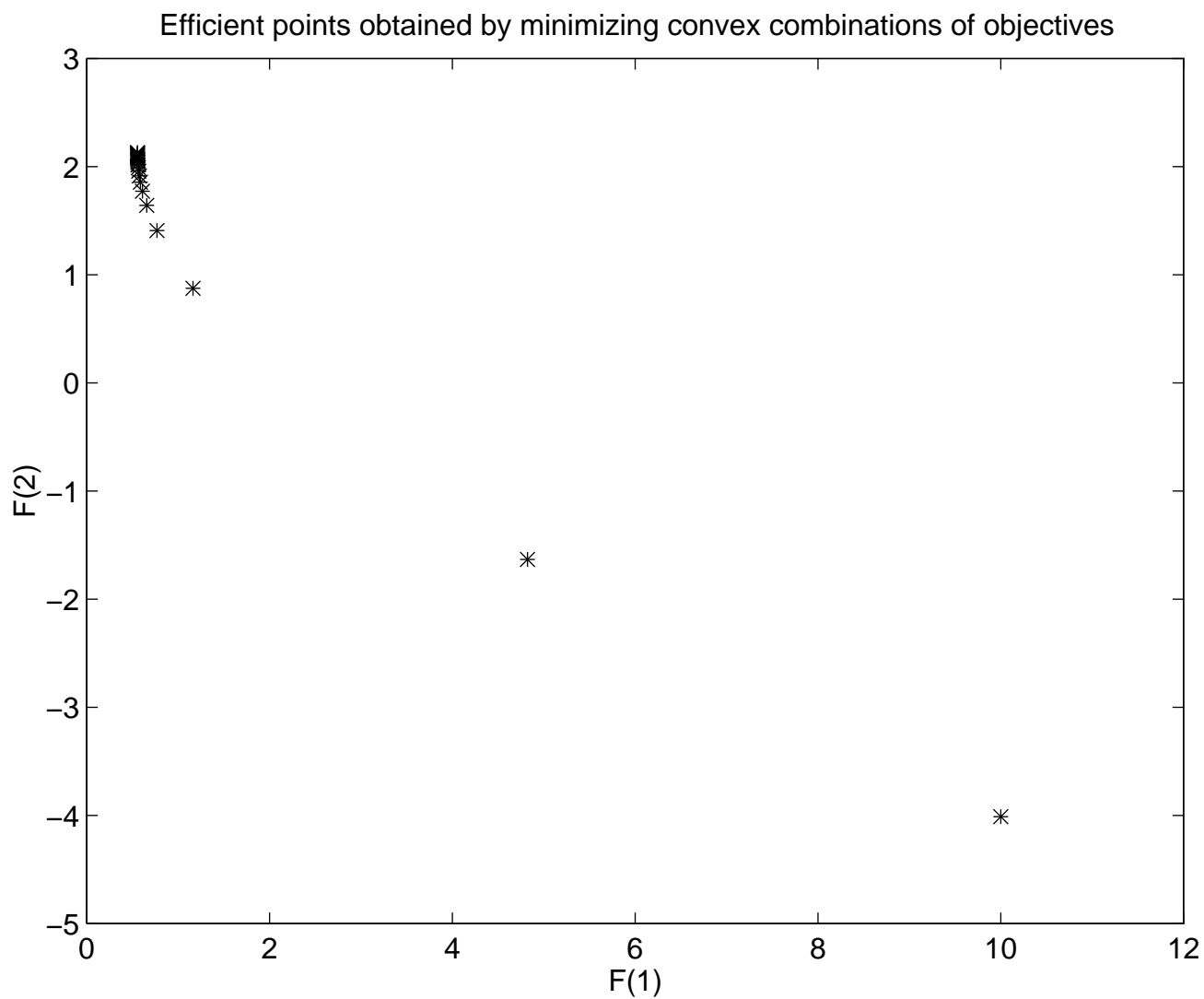
Figure 9: Pareto optimal vectors in the objective space using the method of linear combinations on the problem with f_1 scaled by 10