A Generalization of the Construction of Test Problems for Nonconvex Optimization

KHOSROW MOSHIRVAZIRI
Department of Information Systems, California State University, Long Beach, CA 90840, U.S.A.

(Received: 16 April 1992; accepted: 22 September 1993)

Abstract. In this paper we adopt and generalize the basic idea of the method presented in [3] and [4] to construct test problems that involve arbitrary, not necessarily quadratic, concave functions, for both Concave Minimization and Reverse Convex Programs

Key words. Reverse convex programs, concave programs, test problems, nonconvex problems, global optimization.

1. Introduction

To evaluate an algorithm and measure its efficiency, complexity, and applicability, often one needs to have a variety of test problems with known global solutions. This is particularly important for the class of *Nonconvex Optimization* algorithms and, especially, *Concave Minimization* and *Reverse Convex Programming* algorithms, where the problem usually has many local solutions. While there seems to be a wide interest among researchers in developing algorithms for the latter class of problems, there is still much work to be done in the area of generating test problems for the true evaluation of these proposed algorithms. To this date, only a few authors have published in the area of test problems construction for global optimization algorithms [5]. The purpose of this paper is to adopt the basic idea of the general method presented in [4] (also see [3, 6, 7, 1]) to construct test problems that involve arbitrary, not necessarily quadratic, concave functions, for both Concave Minimization and Reverse Convex programs. The methods presented in [6, 1] require the solution of n linear programs while in [7] a convex programming problem has to be solved. The major computational requirement in [4, 3] and essentially in this paper, involve the solution of a linear system with a unique solution. In addition, these methods require some computational effort to obtain a nondegenerate vertex of the associated convex polyhedron.

2. Case I – Reverse Convex Problem

The global optimization problem, $Minimize\{c^{\top}x \mid x \in P,\ g(x) \leq 0\}$ is known as Linear Reverse Convex Program, LRCP, and the constraint $g(x) \leq 0$ is called reverse convex constraint if g is concave on R^n. In this formulation P is a nonempty convex polyhedron in R^n and c is an n-vector. Further detail on LRCP can be found in [1] and references herein. We now proceed with our procedure of generating test problem for LRCP. Let P be a bounded convex polyhedron in R^n. In addition, assume a nondegenerate vertex x^0 of P is available. (There are several computational procedures for finding such a vertex.) Select a point x^* on the edge $[x^0, x^i]$ of P, where x^i is a selected neighboring vertex of x^0. By the nondegeneracy assumption on x^0, there are exactly n such neighbors to x^0, which are denoted by x^j, $j = 1, \ldots, n$. Thus, $x^* = \lambda x^0 + (1 - \lambda)x^i$ for $0 < \lambda < 1$. Without loss of generality, assume $i = 1$. Additionally, let $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top$ be a prespecified constant vector such that $\alpha_1 = \lambda < 1$ and $\alpha_j > 1$ for $j = 2, 3, \ldots, n$. Then, for $j = 1, \ldots, n$, set $y^j = x^0 + \alpha_j(x^j - x^0)$. Clearly, by construction, $x^* = y^1$. Furthermore, assume that $h(x)$ is a concave function on R^N or on an arbitrary subset of R^n containing P. For instance, $h(x)$ may be chosen to include exponential, logarithmic, trigonometric, or any other type of concave terms. Next, define

$$g(x) = h(x) + c^\top x + g^0 \tag{1}$$

where the augmented linear term, $c^\top x$, which is indeed a control parameter to this proposed model, and the constant term, g^0, are to be determined by imposing the condition that $g(x)$ must vanish at all the points x^0 and y^j, $j = 1, \ldots, n$. That is,

$$g(x^0) = 0$$
$$g(y^j) = 0\,, \quad j = 1, \ldots, n\,.$$

Clearly, the addition of a linear, concave term to $h(x)$ does not disturb the concavity of the resultant function. Next, $g(x^0)$ is subtracted from the last n equations to yield the following set of linear equations:

$$g(y^j) - g(x^0) = 0\,, \quad j = 1, \ldots, n\,.$$

Then, by expanding the terms using equation (1) we get:

$$c^\top(y^j - x^0) = h(x^0) - h(y^j)\,, \quad j = 1, \ldots, n\,.$$

But, for $j = 1, \ldots, n$, $y^j = x^0 + \alpha_j(x^j - x^0)$ and hence $(x^j - x^0) = (y^j - x^0)/\alpha_j$. Recall that with choice of numbers α_j with

$$0 < \alpha_1 < 1 \quad \text{and} \quad \alpha_j > 1\,, \quad j = 2, \ldots, n\,.$$

we get:

$$c^\top(x^j - x^0) = (h(x^0) - h(y^j))/\alpha_j \equiv \beta_j\,.$$

Thus, we have a linear system, n equations in n unknowns, of the form

$$Bc = \beta \ , \tag{2}$$

where B is an $n \times n$ matrix with its i-th row given by $(x^i - x^0)^\top$ and β an n-vector with the components β_j, as defined earlier. By the nondegeneracy and boundedness assumptions on x^0 and P, respectively, the system (2) has a unique solution for c. Once c is known the value of the constant term, g^0, can easily be found as follows:

$$g(x^0) = h(x^0) + c^\top x^0 + g^0 = 0 \implies g^0 = -h(x^0) - c^\top x^0 \ .$$

Therefore, $g(x) = h(x) + c^\top x + g^0 \leq 0$ is the desired nontrivial reverse convex constraint. Next, we attempt to construct an objective hyperplane, $H = \{u \in R^n \mid u^\top x = K\}$ for some constant K, in such a way that the prespecified point x^* on the edge connecting x^0 to x^1 becomes a unique global solution for the reverse convex program, $Minimize\{u^\top x \mid x \in P,\ g(x) \leq 0\}$. To determine H, let us define vectors z^j, $j = 1, \ldots, n$ as follows:

$$z^1 = (x^1 - x^0) \ ,$$

$$z^j = (y^j - x^0) \ , \quad j = 2, \ldots, n \ .$$

Furthermore, let D be an $n \times n$ matrix with vector z^j as its j-th volume and e be a vector of all ones.

$$D = [z^1, z^2, \ldots, z^n] \quad \text{and} \quad e^\top = (1, 1, \ldots, 1) \ .$$

Then the hyperplane H below will have the desired property.

$$H = \{x \mid u^\top (x - x^0) = 1\} \ , \quad u^\top = e^\top D^{-1} \quad \text{and} \tag{3}$$

$$x^* = Argmin\{u^\top x \mid x \in P,\ g(x) \leq 0\} \ . \tag{4}$$

Clearly, the choice of $h(x) = \frac{1}{2} x^\top Q x$, with appropriate matrix Q, can be considered as a special case of the above. Also note that $h(x)$ can be chosen quite arbitrarily as long as it is concave on R^n. Through the choice of the augmented linear term, we correct the resulting function $g(x)$ to satisfy our requirements. Therefore, a wide variety of nontrivial and complex forms of $g(x)$ may be generated.

If P is an unbounded convex polyhedron, then the insertion of the constraint $e^\top x \leq M$, for a large value of M, into the constraint set makes P bounded. Alternatively, the boundedness restriction imposed on P can be relaxed with a minor modification. Let P be an unbounded convex polyhedron in R^n and x^0 be a nondegenerate vertex of P with less than n neighboring vertices. Without loss of generality, assume that η^j, for $j = 2, \ldots, k+1$, $(k \leq n)$ are the directions of the extreme rays leading to x^0 and that x^0 has exactly $(n - k)$ neighboring vertices.

Then the points y^j are determined as follows:

$$y^1 = x^0 + \alpha_1(x^1 - x^0) \quad 0 < \alpha_1 < 1\,, \quad \text{i.e. for } j = 1\,.$$
$$y^j = x^0 + \alpha_j \cdot \eta^j \quad \alpha_j > 0\,, \quad j = 2, \ldots, k+1\,.$$
$$y^j = x^0 + \alpha_j(x^j - x^0) \quad \alpha_j > 1\,, \quad j = k+2, \ldots, n\,.$$

In addition, for $j = 2, \ldots, k+1$, we let $x^j = x^0 + 2\alpha_j \cdot \eta^j$ on the unbounded edges. Note that the generated test problem has the property that x^*, its global solution, lies on an edge leading to x^0. Thus, its detection becomes an easy task for edge-searching algorithms which initiate their searching process from x^0. This, of course, may not be the case for other methods of solving reverse convex programs. Nevertheless, one can modify the test generation method presented here to create more interesting problems. The method described below resets x^0 to a vertex arbitrarily far (in the sense of the minimum number of pivots needed to reach x^* when starting from x^0) from x^*. Thus, the detection of x^* is a nontrivial task for edge-search algorithms. The basic idea behind the method is that one should slice the polytope P in such a way that x^* remains intact and x^0 is chopped out by each slicing plane.

2.1. SLICING PROCEDURE

Let P, x^0, and y^j, $j = 1, \ldots, n$ be given as before. If P is unbounded then the points y^j's and x^j's on the unbounded edges have to be determined according to the procedure described above. Next, assume that $\hat{H}$ is a translation of the plane H, given by equation (3), to the point x^*. Let p^i, $i = 1, \ldots, n$, denote the points where the extreme rays emanating from x^0 intersect $\hat{H}$.

$$p^1 = y^1 \tag{5}$$

$$p^i = x^0 + \theta(y^i - x^0)\,, \quad i = 2, \ldots, n\,. \tag{6}$$

$$\theta = \|y^1 - x^0\| / \|x^1 - x^0\| = \alpha_1\,. \tag{7}$$

Similarly, let us define the points q^i, $i = 1, \ldots, n$ on the extreme rays as follows:

$$q^i = x^0 + \tfrac{1}{2}(p^i - x^0)\,, \quad i = 1, \ldots, n\,.$$

That is, q^i's are the points of intersection with H when it is translated to the midpoint of the line segment (x^0, x^*). Next, let the integer $\hat{m} = \lfloor m/2 \rfloor + 1$ where m is the number of constraints used to define P. Then the points v^i_j, $j = 1, \ldots, \hat{m} - 1$ and for $i = 1, \ldots, n$ are selected as follows:

$$v_j^1 = q^1 + \left(1 - \frac{j}{\hat{m}}\right)(y^1 - q^1)$$

$$v_j^i = x^0 + \frac{j}{\hat{m}}(q^i - x^0)\,, \quad i = 2, \ldots, n\,.$$

Now, if we construct the planes:

$$H_j = \{x \mid e^\top D_j^{-1}(x - x^0) = 1\}\,, \quad j = 1, \ldots, (\hat{m} - 1)\,. \tag{8}$$

where D_j is an $n \times n$ matrix with $(v_j^i - x^0)$ as its i-th column, that is

$$D_j = [v_j^1 - x^0, v_j^2 - x^0, \ldots, v_j^n - x^0]$$

then these planes slice P at x^0. It is easy to see that each plane H_j deletes one vertex from P and creates at least $(n - 1)$ new vertices. This implies that after $(\hat{m} - 1)$ slicing operations, the point x^* will be accessible from $\hat{x}^0$ via at least $\hat{m}$ number of edge-searches. Where $\hat{x}^0$ is the solution to the associated linear program with $(\hat{m} - 1)$ number of slices added to P, i.e.,

$$\hat{x}^0 = Argmin\{e^\top D^{-1}x \mid x \in P, e^\top D_j^{-1}(x - x^0) \geq 1, j = 1, \ldots, \hat{m} - 1\}\,.$$

Note that the columns of D_j may be written as:

$$(v_j^1 - x^0) = \theta\left(1 - \frac{j}{2\hat{m}}\right)(x^1 - x^0) \tag{9}$$

$$(v_j^i - x^0) = \frac{\theta j}{2\hat{m}}(y^i - x^0)\,, \quad i = 2, \ldots, n\,. \tag{10}$$

and for $j = 1, \ldots, (\hat{m} - 1)$.

Therefore, given x^0, x^1 and $\hat{m}$, the slicing hyperplanes are systematically generated and added to P. To derive the equations of the slicing planes, let constants γ_j^i, for $j = 1, \ldots, (\hat{m} - 1)$ be defined as:

$$\gamma_j^1 = \theta(1 - j/2\hat{m})\,, \tag{11}$$

$$\gamma_j^i = \theta j/2\hat{m}\,, \quad i = 2, \ldots, n\,. \tag{12}$$

Then, $e^\top D_j^{-1} = \delta_j^\top D^{-1}$ where

$$\delta_j^\top = (1/\gamma_j^1, 1/\gamma_j^2, \ldots, 1/\gamma_j^n)\,, \quad j = 1, \ldots, (\hat{m} - 1)\,.$$

Thus, the slicing planes H_j's may be written as:

$$H_j = \{x \mid \delta_j^\top D^{-1}(x - x^0) = 1\}\,, \quad j = 1, \ldots, (\hat{m} - 1)\,. \tag{13}$$

Once D^{-1} is computed, the slicing planes H_j's are easily generated according to the equation (13) above. Clearly, by construction, the point x^* remains globally optimal for the generated test problem.

3. Case II – Concave Minimization Problem

Let polytope P and vertex x^0 be given as before and define $V(P)$ to be the set of vertices of P. Furthermore, let C be a *cone* with its main vertex at x^0 and its rays emanating from x^0 to its adjacent vertices. We then truncate the cone produced in this way by a hyperplane which does not cut P, possibly by making use of one of the nonbinding constraints of P at x^0. Otherwise, we can pick $d \in R^n$ such that x^0 is a unique solution to the linear program

$$\mathit{Minimize}\{d^{\top}x \mid x \in P\} . \tag{14}$$

We then solve

$$\mathit{Maximize}\{d^{\top}x \mid x \in P\} \tag{15}$$

to find the vertex $\bar{x} \in P$, and next truncate the cone by constructing the hyperplane $\bar{H}$

$$\bar{H} = \{x \mid d^{\top}x = d^{\top}\bar{x}\}$$

to generate vertices $s^1, s^2, \ldots, s^n$. To determine d, and subsequently $s^1, s^2, \ldots, s^n$ first we construct a *polar cone* C^* of C, with its generators taken as the normalized gradients of the binding constraints at x^0. Then we select d such that $-d \in \mathit{int}\, C^*$. This guarantees that x^0 will uniquely solve the linear program (14). Of course, other methods that can use the results of the optimal simplex tableau corresponding to x^0 may be used as well. Also, the hyperplane H given by (3) of Case I can be used. Let S denote a simplex with its main vertex at x^0 and with the vertices $s^1, \ldots, s^n$ containing P, $S \supseteq P$. In addition, let $h(x)$ be a known concave function on R^n or on a subset of R^n containing P. Clearly, a variety of complicated concave functions may be selected for $h(x)$. Then, similar to the last case, set

$$f(x) = h(x) + c^{\top}x + f^0$$

where the control term, $c^{\top}x$, the constant term f^0, and vector c are to be determined by imposing the condition that $f(x)$ must vanish at x^0 and at all the other vertices of the simplex S, namely s^j, $j = 1, \ldots, n$. That is,

$$f(x^0) = 0$$
$$f(s^j) = 0\,, \quad j = 1, \ldots, n\,.$$

By subtracting the first equation from each of the last n equations and then by substituting and expanding the terms we obtain

$$f(s^j) - f(x^0) = 0\,, \quad j = 1, \ldots, n$$
$$c^\top(s^j - x^0) = h(x^0) - h(s^j) \equiv \beta_j\,, \quad j = 1, \ldots, n\,.$$

Consequently, this reduces to the system $Bc = \beta$ with a unique solution for control vector c. The value of f^0, as in Case I, can be obtained as

$$f^0 = -h(x^0) - c^\top x^0\,.$$

By way of construction, all the vertices of $V(P)\backslash\{x^0\}$ are strictly interior to the surface $\{x \in R^n \mid f(x) = 0, x \in P\}$. This simply implies that for each vertex $x^j \in V(P)$, $j \neq 0$, we have $f(x^j) > 0$. This in turn implies that

$$0 \overset{\text{def}}{=} f(x^0) \leqslant f(x^j)\,, \quad \forall x^j \in V(P)\,, \quad j = 1, \ldots, n$$

which leads to the conclusion that

$$x^0 = Argmin\{f(x) \mid x \in V(P)\}\,.$$

Then, by the virtue of the convexity of P and concavity of f, it follows that

$$x^0 = Argmin\{f(x) \mid x \in P\}\,.$$

That is, x^0, a prespecified nondegenerate vertex of P, is a global minimum for the constructed concave function $f(x)$ on P.

The simple theorem below, the proof of which has already been given above, summarizes the main results of this section.

THEOREM 1. *Let a polytope P, a nondegenerate vertex x^0 of P, a point x^*, and an arbitrary concave function $h(x)$ be given with the desired properties as described in this section. Furthermore, assume that the control terms, $c^\top x + g^0$ and $c^\top x + f^0$, are computed according to the procedures of Case I and Case II, respectively. Then, x^* will globally minimize the reverse convex program with the reverse constraint given by $g(x) = h(x) + c^\top x + g^0 \leqslant 0$ on P. Similarly, x^0 will globally minimize the concave function $f(x) = h(x) + c^\top x + f^0$ on P.*

4. Numerical Examples

In this section we present two examples for each of our two cases. Examples 1 and 2 for the concave programming followed by Examples 3 and 4 for the reverse convex programming. Also note that $\log(x)$ denotes the natural logarithm of x throughout this section.

4.1. EXAMPLE 1

This problem is taken from Falk–Hoffman [2]. The original problem presented in [2] is a Concave Program with the objective function given by

$$f(x) = -(x_1 - 1)^2 - x_2^2 - (x_3 - 1)^2 \tag{16}$$

and P, the associated polytope is defined as the intersection of the following linear constraints in R^3.

$$\begin{aligned} x_1 + x_2 - x_3 &\leq 1 \\ x_1 - x_2 + x_3 &\geq 1 \\ 12x_1 + 5x_2 + 12x_3 &\leq 34.8 \\ 12x_1 + 12x_2 + 7x_3 &\leq 29.1 \\ 6x_1 - x_2 - x_3 &\geq 4.1 \end{aligned}$$

$$x_1 \geq 0\,, \quad x_2 \geq 0\,, \quad x_3 \geq 0\,.$$

Let $V(P) = \{v^0, v^1, \ldots, v^9\}$ denote the set of vertices of P. Where,

$$\begin{aligned} v^0 &= (0.72857142857, 0, 0.27142857143)^\top \\ v^1 &= (0.986713236, 0.9034964, 0.9167832)^\top \\ v^2 &= (1.0703702, 0, 2.3222222)^\top \\ v^3 &= (4.41998, 0, -3.420004)^\top \\ v^4 &= (1.9, 0, 0.9)^\top \\ v^5 &= (1, 0.9, 0.9)^\top \\ v^6 &= (1, 0, 0)^\top \\ v^7 &= (1.033332, 0.4, 1.7)^\top \\ v^8 &= (1.76, 0, 1.14)^\top \\ v^9 &= (1, 0, 1.9)^\top\,. \end{aligned}$$

The global minimum is attained at the vertex $v^6 = (1, 0, 0)^\top$ of P with the objective value of $f(v^6) = -1$. We select v^6 as our initial nondegenerate vertex and will implement the method of Case II to obtain an objective function, more complicated in form than (16), that attains its global minimum on P at v^6.

Starting at v^6, first we construct a cone C with its main vertex at v^6, by extending the rays emanating from v^6 to v^0, v^4, and v^5, its three neighboring vertices. Clearly, $P \subset C$. Let $D = [v^0 - v^6, v^4 - v^6, v^5 - v^6]$ and $d^\top = e^\top D^{-1}$. Then, the hyperplane H given by

$$H = \{x \in R^n \mid d^\top (x - v^6) = 1\}$$

passes through the vertices v^0, v^4, v^5. As a result, $v^9 = (1, 0, 1.9)^{\top} = Argmax\{d^{\top}x \mid x \in P\}$. This means that we can truncate C by $\bar{H}$, the translation of H to the point v^9, to construct the simplex S and consequently find s^1, s^2, s^3, the vertices of S. As a result of this process we obtain

$$\bar{H} = \{x \in R^n \mid d^{\top}(x - v^9) = 0\}$$

$$\text{for} \quad d^{\top} = (-1.286549708, -1.286549708, 2.397660819)$$

and vertices of the simplex S, with its main vertex at v^6, are found to be

$$s^1 = (-0.236507936426, 0, 1.23650793643)^{\top}$$
$$s^2 = (5.1000000009, 0, 4.1000000009)^{\top}$$
$$s^3 = (1, 4.1000000009, 4.1000000009)^{\top}.$$

The next step is to select $h(x)$. We can easily take any complex concave form for $h(x)$. However, to serve the aims of this illustration, it suffices to choose a nontrivial form which is also quite different from (16). Thus we select

$$h(x) = 6(1 - e^{-\frac{1}{2}x_1}) - x_2^2 + 18(1 - e^{-\frac{1}{4}x_3}). \tag{17}$$

Given $h(x)$ and v^6, s^1, s^2, s^3, matrix B and vector β for the system $Bc = \beta$ are obtained as

$$B = \begin{bmatrix} -1.23650793643 & 0 & 1.23650793643 \\ 4.10000000090 & 0 & 4.10000000090 \\ 0 & 4.10000000090 & 4.10000000090 \end{bmatrix}$$
$$\beta^{\top} = (-1.67242857509, -14.7123575866, 5.26833638323).$$

Upon solving the system, we obtain

$$c^{\top} = (-1.11791908759, 3.75542090399, -2.47046081080)$$

The value of the constant term, f^0, is found to be

$$f^0 = -h(v^6) - c^{\top}v^6 = -6(1 - e^{-0.5}) - c_1 = -1.24289695413$$

and, as a result, the constructed objective function may be written as

$$\begin{aligned} f(x) = {} & 6(1 - e^{-\frac{1}{2}x_1}) - x_2^2 + 18(1 - e^{-\frac{1}{4}x_3}) \\ & - 1.11791908759x_1 + 3.75542090399x_2 - 2.47046081080x_3 \\ & - 1.24289695413 \end{aligned}$$

which attains its global minimum on P at the same vertex v^6 with $f(v^6) = 0$.

4.2. EXAMPLE 2

As an alternative and slightly more complex form of $h(x)$ than our first choice, we consider the following function:

$$h(x) = \log(1 + x_1) - e^{-x_1} - x_1^2 + x_2^{3/4} - x_2^4 + 3(1 - e^{-0.36x_3})$$

where $h(x)$ is concave on P. Taking v^6 as vertex x^0 in the procedure of Case II and implementing the process, we find, $\bar{H}$, the plane truncating the cone, as in Example 1, matrix B, and the vector β as below:

$$\bar{H} = u^{\mathsf{T}}x \quad \text{for} \quad u^{\mathsf{T}} = (-1.286549708, -1.286549708, 2.397660819)$$

$$B = \begin{bmatrix} -1.23650793643 & 0 & 1.23650793643 \\ 4.10000000090 & 0 & 4.10000000090 \\ 0 & 4.10000000090 & 4.10000000090 \end{bmatrix}$$

$$\beta^{\mathsf{T}} = (-0.159927889068, 21.2187259116, 277.380454328)\,.$$

Thus, the control term, c, is found to be:

$$c^{\mathsf{T}} = (2.65231867294, 65.1307890046, 2.52298032875)\,.$$

The value of the constant term $f^0 = -h(v^6) - c^{\mathsf{T}}v^6$ is -1.97758641233. Note that matrix B and plane H^0 are the same as those of example 1. Therefore, the second alternative form for the Falk–Hoffman objective function may be given as

$$\begin{aligned} f(x) &= \log(1 + x_1) - e^{-x_1} - x_1^2 + x_2^{3/4} - x_2^4 + 3(1 - e^{-0.36x_3}) \\ &\quad + 2.65231867294x_1 + 65.1307890046x_2 + 2.52298032875x_3 \\ &\quad - 1.97758641233 \end{aligned}$$

where, once again, the global minimum of $f(x)$ on P is attained at v^6. Clearly, by construction, the optimal value of $f(x)$, which is always attained at the selected vertex, is zero for every test problem generated by this method.

4.3. EXAMPLE 3

Consider the Falk–Hoffman polytope given in Example 1. Let x^0 and its neighboring vertices x^1, x^2, and x^3, as defined in the procedure of Case I, be given as follows:

$$\begin{aligned} x^0 &= v^6 = (1, 0, 0)^{\mathsf{T}} \\ x^1 &= v^5 = (1, 0.9, 0.9)^{\mathsf{T}} \\ x^2 &= v^4 = (1.9, 0, 0.9)^{\mathsf{T}} \\ x^3 &= v^0 = (\tfrac{51}{70}, 0, \tfrac{19}{70})^{\mathsf{T}}\,. \end{aligned}$$

Also, let us choose $m = 3$, $\alpha = (\frac{1}{3}, 10, \frac{70}{19})^{\mathrm{T}}$. (Recall that we must have $\alpha_1 < 1$, and $\alpha_i > 1$, $i = 2, \ldots, n$), and an arbitrary concave function $h(x)$

$$h(x) = 6(1 - e^{-\frac{1}{2}x_1}) - x_2^2 + 18(1 - e^{-\frac{1}{4}x_3}) . \tag{18}$$

Thus, we find $y^1 = (1, 0.3, 0.3)^{\mathrm{T}}$, $y^2 = (10, 0, 9)^{\mathrm{T}}$, and $y^3 = (0, 0, 1)^{\mathrm{T}}$. Likewise, the matrices B, B^{-1}, D, and D^{-1} are found accordingly.

$$B = \begin{bmatrix} 0 & 0.9 & 0.9 \\ 0.9 & 0 & 0.9 \\ -19/70 & 0 & 19/70 \end{bmatrix}$$

$$B^{-1} = \begin{bmatrix} 0 & 5/9 & -19/35 \\ 10/9 & -5/9 & -19/35 \\ 0 & 5/9 & 19/35 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 9 & -1 \\ 0.9 & 0 & 0 \\ 0.9 & 9 & 1 \end{bmatrix}$$

$$D^{-1} = \frac{1}{18} \begin{bmatrix} 0 & 20 & 0 \\ 1 & -1 & 1 \\ -9 & -9 & 9 \end{bmatrix}$$

The vector β and c which solves $Bc = \beta$ and g^0 are found as:

$$g^{\mathrm{T}} = (-3.63185173825815, -1.97015702341677, -0.4399232485259963)$$

$$c^{\mathrm{T}} = (-0.2841467481807287, -2.130474209115588, -1.904916611171242)$$

$$g^0 = -2.076669293543471 .$$

Consequently, the vectors δ_j, $j = 1, \ldots, (m - 1)$, the plane H (to be minimized) and the two slicing planes, H_1 and H_2 are found to be:

$$\delta_1 = (18/5, 18, 18)^{\mathrm{T}} , \quad \delta_2 = (9/2, 9, 9)^{\mathrm{T}}$$

$$H = \{x \mid -4x_1 + 5x_2 + 5x_3 = 5\} \tag{19}$$

$$H_1 = \{x \mid -8x_1 - 6x_2 + 10x_3 = -7\} \tag{20}$$

$$H_2 = \{x \mid -4x_1 + 0x_2 + 5x_3 = -3\} . \tag{21}$$

Note that for simplicity of presentation the coefficients in H and H_j's above may be scaled (here they were scaled by factor of 9). Therefore, the generated reverse convex test problem may now be given as:

Minimize $\quad -4x_1 + 5x_2 + 5x_3$

Subject to:

$$\begin{aligned}
x_1 + x_2 - x_3 &\leqslant 1 \\
x_1 - x_2 + x_3 &\geqslant 1 \\
12x_1 + 5x_2 + 12x_3 &\leqslant 34.8 \\
12x_1 + 12x_2 + 7x_3 &\leqslant 29.1 \\
6x_1 - x_2 - x_3 &\geqslant 4.1 \\
-8x_1 - 6x_2 + 10x_3 &\geqslant -7 \\
-4x_1 + 5x_3 &\geqslant -3 \\
x_1 \geqslant 0, x_2 \geqslant 0, x_3 &\geqslant 0 . \\
g(x) &\leqslant 0
\end{aligned} \tag{22}$$

where,

$$\begin{aligned}
g(x) = {} & 6(1 - e^{-\frac{1}{2}x_1}) - x_2^2 + 18(1 - e^{-\frac{1}{4}x_3}) - 0.2841467481807287x_1 \\
& - 2.130474209115588x_2 - 1.904916611171242x_3 \\
& - 2.076669293543471 .
\end{aligned}$$

The global minimum is attained at the point $x^* = (1, 0.3, 0.3)^T$ on the edge between x^0 and x^1, as expected and with the objective value of $z^* = -1$. Furthermore,

$$\hat{x}^0 = (1.8920454545, 0, 0.9136363636)$$

is an optimal solution of the associated linear program. Clearly, $\hat{x}^0$ is not on any edge of P that leads to x^*. In fact, several pivot operations are needed to reach x^1 starting from $\hat{x}^0$. The following alternative solution to the linear program was reported by the referee.

$$\hat{x}_1^0 = (0.8888888889, 0, 0.1111111111) .$$

4.4. EXAMPLE 4

This example is a counterpart of Example 2 for the concave test generating problem. That is, the arbitrary concave function $h(x)$ is chosen to be a slightly more complex than that in the Example 3.

$$h(x) = \log(1 + x_1) - e^{-x_1} - x_1^2 + x_2^{3/4} - x_2^4 + 3(1 - e^{-0.36x_3}) .$$

Assume all other parameters of the previous example remain unchanged except for the choice of function $h(x)$. An implementation of the procedure of Case I, omitting the intermediate computations, results in the following choices for c and g^0.

$$c^{\mathrm{T}} = (5.515568838522362, -7.281789746626642, 4.933865556123956)$$
$$g^0 = -4.840836577910864\,.$$

Of course, the determination of the slicing planes, the predetermined global solution, x^*, and the objective function remain unaffected. Thus, the new generated test problem is the same as that of Example 3, except $g(x)$ in the constraint (22) is replaced by

$$\begin{aligned} g(x) = {} & \log(1 + x_1) - e^{-x_1} - x_1^2 + x_2^{3/4} - x_2^4 + 3(1 - e^{-0.36x_3}) \\ & + 5.515568838522362x_1 - 7.281789746626642x_2 \\ & + 4.933865556123956x_3 - 4.840836577910864 \end{aligned}$$

where, once again the global minimum is attained at the point x^*, as expected.

5. Concluding Remarks

In this paper we considered a method for generating test problems for linear reverse convex programs and concave programming. The nonconvex functions involved in both cases are not restricted to possess specific structure. For instance, they are not necessarily quadratic as is the case in [3, 4, 6, 7]. In addition, a slicing procedure was presented in order to generate test problems that are not trivially solved by algorithms that are based on searching the edges of the associated polytope. Also, the constraint matrix defining P may be made sparse or dense by selecting small or large value for $\hat{m}$, the number of slicing planes, respectively. And finally, it is quite an easy task to implement the method. Using the proposed method and taking P as a unit hypercube, or randomly generated polytope, we constructed test problems of 3 to over 50 variables. These test problems were not solvable by general nonlinear programming codes such as GRG2 and MINOS. Of course, this was expected due to nonconvexity of the feasible regions and the functions involved. Furthermore, the generated test problems proved to be hard for a few specialized global optimization codes, (e.g. algorithms based on Edge-Searching and Cutting Plane approaches) that was available to us.

Acknowledgement

The author would like to thank the Editor and the referees for their patient reading of this manuscript and for their constructive comments. The author also wishes to express his appreciation to Prof. S. E. Jacobsen for his encouragement and insights. This work was completed while the author was a Post Doctoral fellow at the University of California, Los Angeles, California.

References

1. BenSaad, S. and Jacobsen, S. E. (1990), A Level Set Algorithm for a Class of Reverse Convex Programs, *Annals of Operations Research* **25**, 19–42.
2. Falk, J. E. and Hoffman, K. L. (1976), A Successive Underestimation Method for Concave Minimization Problems, *Mathematics of Operations Research* **1**(3), 251–259.
3. Kalantari, B. and Rosen, J. B. (1986), Construction of Large-Scale Global Minimum Concave Quadratic Test Problems, *Journal of Optimization Theory and Applications* **48**, 303–313.
4. Pardalos, P. M. (1987), Generation of Large-Scale Quadratic Programs for Use as Global Optimization Test Problems, *ACM Transaction on Mathematical Software* **13**(2), 143–147.
5. Floudas, C. A. and Pardalos, P. M. (1990), A Collection of Test Problems for Constrained Global Optimization Algorithms, *Springer Verlag Lecture Notes in Computer Science* 455.
6. Sung, Y. and Rosen, J. B. (1982), Global Minimum Test Problem Construction, *Mathematical Programming* **24**, 353–355.
7. Rosen, J. B. (1983), Global Minimization of a Linearly Constrained Concave Function by Partition of Feasible Domain, *Mathematics of Operations Research* **8**(2), 215–230.