Mathematical Programming 1 (1971) 113–116. North-Holland Publishing Company

NOTE

AN EXAMPLE OF AN ILL-CONDITIONED NLP PROBLEM

T.B. CRABILL, J.P. EVANS and F.J. GOULD
University of North Carolina, Chapel Hill, North Carolina, U.S.A.

Received 8 December 1970
Revised manuscript received 2 March 1971

1. Introduction

In this note a new nonlinear programming test problem is presented. Although the problem is not convex, all derivatives of all of the functions in the problem exist, the objective function is linear, the constraint set compact, and hence it might appear, at least by many conventional measures, that the problem is reasonably well behaved. However, it cannot be solved by a number of the existing algorithms. In section 2 the problem is presented along with computational experience on the performance of two codes which represent the state of the art in gradient and penalty function techniques, respectively. In section 3 the test problem is discussed and it is shown that in spite of its smoothness properties it is ill behaved in a well defined sense. In particular, the constraint mapping associated with this problem is not upper semicontinuous. Upper semicontinuity is a property which in a previous study [2] has been shown to be or theoretic interest. The purpose of this note is to demonstrate the numerical importance of this property, and to indicate that when upper semicontinuity fails, certain kinds of algorithms will not perform as desired.

2. Numerical results

The test problem is

$$\max f(x_1, x_2) = x_1 + x_2, \qquad \text{subject to}$$

$$g(x_1, x_2) = \exp[(-x_1^2 - x_2^2 - 4)/4] - \exp[(-x_1^2 - x_2^2)/2] = 0.$$

A cross-section of the graph of $g(.)$ is shown in fig. 1. It is readily verified that the constraint set for this problem is the circle with center at the origin and radius 2, and that the unique solution to the problem is $(\sqrt{2}, \sqrt{2})$.

The problem was run on two codes, GRG and SUMT*. In order to conform with the GRG format the following redundant bounds were added

$$-20 \leqslant x_1 \leqslant 20$$

$$-20 \leqslant x_2 \leqslant 20,$$

and the problem with the bounds was run on both codes, using (2, 0) as the starting point. Both codes gave false indications of optimality at (20, 20).

In another test the constraint was modified to $g(x_1, x_2) \leqslant 0$ and the problem was again run on both codes. In order to conform with the SUMT treatment of inequality constraints, the origin was used as the starting point for these runs. The correct solution was reached with both codes.

* The GRG code was developed by Abadie and Carpentier (see [1]) at Electricité de France. The SUMT code was developed by Fiacco and McCormick (see [3]) at Research Analysis Corporation.

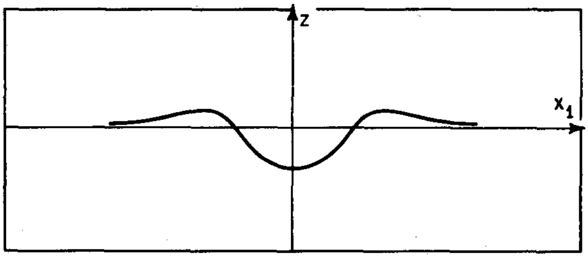

Fig. 1.

3. Interpretation

Let $g(x)$ be a continuous function from R^n to R^m, and define $S_b = \{x \in R^n : g(x) \leqslant b\}$ where $b \in R^m$. The sets S_b permit us to define the constraint mapping $S : R^m \rightarrow P(R^n)$; that is, the image of b under the mapping S is the set S_b. If S_0 is compact, as in the test problem above, a result in [2] establishes that the constraint mapping is upper semi-continuous at 0 if and only if there is a $\hat{b} > 0, \hat{b} \in R^m$ such that $S_{\hat{b}}$ is compact. The pathology of the problem herein presented is that this property fails; a small increase in the right hand side of the constraint leads to a discontinuous expansion of the constraint set. The set S_0 is compact but the set S_b is unbounded for each $b > 0$. Thus, in terms of finite arithmetic $g(x_1, x_2)$ will appear to be zero for certain large values of x_1 and x_2. Also, in the equality constraint case, the computer test for feasibility is usually in the nature of testing whether or not $|g(x_1, x_2)| \leqslant \epsilon_0$ for some small ϵ_0. Again it is seen that for this problem such a feasibility test can be satisfied by values of x_1, x_2 far from the true constraint set $g(x_1, x_2) = 0$. This statement will be true regardless of how small ϵ_0 is chosen. Thus, this pathology is not simply the usual kind of example where Newton's method might fail. It is, rather, an example to illustrate that if the constraint map is not upper semi-continuous then a "very incorrect" computer indication of optimality can be obtained.

In the case of the constraint $g(x_1, x_2) < 0$, the GRG code works because the parallelotope $-20 \leqslant x_1 \leqslant 20$, $-20 \leqslant x_2 \leqslant 20$ is sufficiently small that evaluating g on the edges does not give a computer zero. If the parallelotope were chosen quite large (say $|x_1| \leqslant 100$, $|x_2| \leqslant 100$) then the value of g on the edges would be a computer zero and the code would fail. Because the SUMT method is inside out, it will generally be successful on an inequality constrained problem, even though it is not upper semi-continuous. However, penalty methods which are not restricted to the interior of the constraint set could generally be expected to fail.

Acknowledgement

The authors are indebted to Electricité de France and to the Research Analysis Corporation for the use of GRG and SUMT, respectively, in this research. This research was partially supported by a grant from the

Office of Naval Research, contract number N00014-67-A-0321-0003 (NR047-095).

References

[1] J.Abadie, "Application of the GRG Algorithm to Optimal Control Problems," in: *Integer and Nonlinear Programming,* ed. J.Abadie (North-Holland Publishing Comp., Amsterdam, 1970).

[2] J.P. Evans and F.J. Gould, "Stability in nonlinear programmin," *Operations Research* 18, No. 1 (1970) 107–118.

[3] A.V. Fiacco and G.P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques* (John Wiley and Sons, Inc., New York, 1968).