# A VARIABLE-COMPLEXITY NORM MAXIMIZATION PROBLEM

by

O. L. Mangasarian
and
T.-H. Shiau

Computer Sciences Technical Report #568

December 1984

UNIVERSITY OF WISCONSIN - MADISON
COMPUTER SCIENCES DEPARTMENT

# A VARIABLE-COMPLEXITY NORM MAXIMIZATION PROBLEM

O. L. Mangasarian and T.-H. Shiau

## ABSTRACT

The decision problem associated with the problem of finding a point with largest norm in a bounded polyhedral set is shown to have a considerable range of complexity depending on the norm employed. For a p-norm with integer $p \geq 1$, the problem is shown to be NP-complete. For the $\infty$-norm, the problem can be solved in polynomial time. The problem of finding an upper bound to the largest norm for any $p \in [1,\infty]$ can be solved in polynomial time by solving a single linear program.

AMS (MOS) Subject Classifications: 03D15, 90C05, 90C30.

Key Words: Optimization, maximum norm, complexity theory, NP-complete.

---

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041. This material is based upon work supported by the National Science Foundation under Grant Nos. MCS-8200632 and DMS-8210950, Mod. 1.

# A VARIABLE-COMPLEXITY NORM MAXIMIZATION PROBLEM

O. L. Mangasarian and T.-H. Shiau

## 1. Introduction

The problem of obtaining bounds for polyhedral sets has received considerable attention in mathematical programming [14, 15, 16, 12, 8, 9]. Part of the significance of this problem stems from the fact that the solution set to a linear program [4, 10] and to a monotone linear complementarity problem [2] is such a polyhedral set. Bounding the solution set to such problems when possible is then of practical interest. In this work we shall consider the polyhedral set $X$ in $R^n$ defined by

(1.1) $$X := \{x \mid x \in R^n,\ Ax \geqq b\}$$

where $A$ is a given $m \times n$ rational matrix and $b$ is a given $m \times 1$ rational vector. We assume throughout this work that $X$ is bounded. It is easy to show that a necessary and sufficient condition for $X$ to be bounded is that

(1.2) $$Y = \{y \mid y \in R^n,\ Ay \geqq 0,\ y \neq 0\} = \phi \ .$$

The problem we wish to consider here is

(1.3) $$\max_{x \in X} \|x\|_p$$

where $\|\cdot\|_p$ denotes the p-norm on $R^n$, $1 \leqq p =$ integer $< \infty$, defined by $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ and $\|x\|_\infty = \max_{1 \leqq i \leqq n} |x_i|$.

We will show that while (1.3) can be solved in polynomial time for $p = \infty$, the decision problem associated with it is NP-complete [6, 11] for integer $p \geqq 1$. Since it is widely believed that <u>no</u> NP-complete problem can be solved in polynomial time (the famous conjecture $P \neq NP$ in computational complexity theory), the difference in the difficulty between $p = \infty$ and all other integer $p \geqq 1$ is enormous. (The standard complexity theory terms used here are defined in Section 4.) In fact we can summarize the complexity situation for our problem (1.3) as shown in Table 1.

---

Sponsored by the United States Army under Contract No. DAAG29-80-C-0041. This material is based upon work supported by the National Science Foundation under Grant Nos. MCS-8200632 and DMS-8210950, Mod. 1.

| Problem | Complexity | Known Method of Solution |
| --- | --- | --- |
| 1. Find an upper bound to $\max_{x \in X} \|x\|_p$ for any $p \in [1,\infty]$ | P (Deterministic polynomial time) | Single linear program |
| 2. $\max_{x \in X} \|x\|_\infty$ | P | $2n$ linear programs |
| 3. $\max_{x \in X} \|x\|_1$ | NP-complete (Nondeterministic polynomial time) | $2^n$ linear programs |
| 4. $\max_{x \in X} \|x\|_p$ Integer $p \geq 2$ | NP-complete | Vertex enumeration |

Table 1. Complexity of $\max_{x \in X} \|x\|_p$ and its method of solution.

We note in passing that the <u>minimization</u> problem $\min_{x \in X} \|x\|_p$ is by contrast a much simpler convex programming problem for $p \in [1,\infty]$. In fact for $p = 1$ and $\infty$ it can be solved by standard linear programming techniques [4, 10] or by a polynomial time algorithm e.g. [7]. For $p = 2$ the problem is a convex quadratic program which can be solved by standard techniques e.g. [2] or by a polynomial time algorithm [3].

In the following sections of this paper we will show how each of the problem of Table 1 is solved and its complexity. Section 2 deals with finding an upper bound to (1.3) for $p \in [1,\infty]$. Section 3 deals with problem (1.3) for $p = 1$ and $\infty$ while Section 4 deals with the cases of integer $p \geq 1$.

## 2. Bounding $\max_{x \in X} \|x\|_p$

It is useful to know that for any $p \in [1,\infty]$, $p$ not necessarily an integer, an upper bound to the solution of the nonconvex problem $\max_{x \in X} \|x\|_p$ can be obtained by solving a single linear program (Theorem 2.1 below). This is a useful result since we show (Section 4) that the problems $\max_{x \in X} \|x\|_p$ for integer $p \geq 1$ are intractable NP-complete problems.

When $X$ is contained in the nonnegative orthant $R^n_+ := \{x \mid x \in R^n,\ x \geqq 0\}$ it is evident that a solution to the 1-norm problem $\max_{x \in X} \|x\|_1$ is easily obtained by the single linear program

(2.1)
$$\max_{x \in X \cap R^n_+} ex$$

where $e$ is a vector of ones. However when $X \not\subset R^n_+$, as may be the case here, solution of $\max_{x \in X} \|x\|_1$ may take $2^n$ linear programs, as shown in Section 3. In fact we will show in Section 4 that the problem $\max_{x \in X} \|x\|_1$ is NP-complete. However, merely obtaining an upper bound to $\max_{x \in X} \|x\|_p$ for any $p \in [1,\infty]$ will take at most a single linear program as shown by the following result.

2.1 Theorem. Let $X$ be nonempty and bounded, let

(2.2)
$$B := (A^TA)^{-1}A^T \quad , \quad d := Bb$$

and let $B_{\cdot j}$ denote the $j^{th}$ column of $B$. Then for any $p \in [1,\infty]$ and any $x \in X$

(2.3)
$$\|x\|_p^p \leqq \max_{1 \leqq j \leqq m} \{\|d\|_p^p,\ \|\gamma B_{\cdot j} + d\|_p^p\}$$

where $\gamma$ is the maximum value of the following solvable linear program

(2.4)
$$\gamma := \max_{x,y}\{ey \mid x \in R^n,\ y \in R^m,\ Ax - y = b,\ y \geqq 0\} \quad .$$

Proof. Note first that the boundedness condition (1.2) implies the linear independence of the columns of $A$ and hence the nonsingularity of $A^TA$. In addition the nonemptiness and boundedness of $X$ implies the solvability of the linear program (2.3). Hence

$$\begin{aligned}
\max_{x \in X} \|x\|_p &= \max_{x,y}\{\|x\|_p \mid x \in R^n,\ y \in R^m,\ Ax - y = b,\ y \geqq 0\} \\
&= \max_{x,y}\{\|x\|_p \mid x = By + d,\ (AB-I)(y+b) = 0,\ y \geqq 0,\ ey \leqq \gamma\} \\
&\leqq \max_{x,y}\{\|x\|_p \mid x = By + d,\ y \geqq 0,\ ey \leqq \gamma\} \\
&= \max_{y}\{\|By + d\|_p \mid y \geqq 0,\ ey \leqq \gamma\} \\
&= \max_{1 \leqq j \leqq m} \{\|d\|_p,\ \|\gamma B_{\cdot j} + d\|_p\}
\end{aligned}$$

where the last equality follows from the fact that the maximum of a convex function on a bounded polyhedral set is attained at a vertex [13, Corollary 32.3.4].

□

Note that if a lower bound to $\max_{x \in X} \|x\|_p$ is also desired, then we have the following.

2.2 Corollary. Under the assumptions of Theorem 2.1 we have that

$$\|B\bar{y} + d\|_p \leqq \max_{x \in X} \|x\|_p$$

where $\bar{y}$ is a solution of the linear program (2.4).

Since by Khachian's result [7] a linear program is solvable in polynomial time in the size of the problem, and since the algebraic operations prescribed in (2.3) can all be performed in polynomial time, the following holds.

2.3 Corollary. The bound (2.3) can be computed in time which is polynomial in the size of $A$ and $b$.

We note that the bound (2.3) of Theorem 2.1 may be sharp as evidenced by the following example.

2.4. Example

$$A = \begin{pmatrix} -2 & 1 \\ 5 & 1 \\ 1 & -4 \end{pmatrix} \qquad b = \begin{pmatrix} -10 \\ -10 \\ -2 \end{pmatrix}$$

For this example it is easy to verify that

$$\max_{x \in X} \|x\|_p = 10 \quad \text{for} \quad p = 1,\ 2 \ \text{and}\ \infty \quad ,\ \gamma = 42$$

$$B = \begin{pmatrix} -.0649 & .1688 & .0260 \\ .0519 & .0649 & -.2208 \end{pmatrix} \quad , \quad d = \begin{pmatrix} -1.0909 \\ -.7273 \end{pmatrix}$$

Computing the bound (2.3) of Theorem 2.1 gives for $p = 1,\ 2$ and $\infty$

$$\max_{1 \leqq j \leqq 3} \{\|d\|_p,\ \|\gamma B_{\cdot j} + d\|_p\} = 10 \quad .$$

## 3. $\max_{x \in X} \|x\|_p$ for $p = \infty$ and 1

It is rather obvious that the problem $\max_{x \in X} \|x\|_\infty$ can be solved by maximizing the absolute value of each component of $x$ separately subject to $x$ in $X$. This leads to the following.

### 3.1 Proposition

The problem $\max_{x \in X} \|x\|_\infty$ can be solved by solving the $2n$ linear programs

$$\max_{1 \leqq i \leqq n} \max\{\pm x_i \mid x \in R^n, Ax \geqq b\} \quad . \tag{3.1}$$

Since each linear program can be solved in polynomial time [7] we have the following.

### 3.2 Corollary

The problem $\max_{x \in X} \|x\|_\infty$ can be solved in time which is polynomial in the size of $A$ and $b$.

Since the problem $\max_{x \in X} \|x\|_1$ is equivalent to $\max_{x \in X} \sum_{i=1}^{n} |x_i|$, its solution can be obtained by solving $2^n$ linear programs as follows.

### 3.3 Proposition

The problem $\max_{x \in X} \|x\|_1$ can be solved by solving the $2^n$ linear programs

$$\max_{v \in V} \max_{x} \{vx \mid x \in R^n, Ax \geqq b\} \tag{3.2}$$

where $V$ is the set of $2^n$ vertices of the cube in $R^n$ defined by

$$\{v \mid v \in R^n, -e \leqq v \leqq e\} \ , \tag{3.3}$$

where $e$ is a vector of ones.

While $2n$ linear programs can be solved in a reasonable amount of time for intermediate-sized problems, solving $2^n$ linear programs is intractable even for $n$ as small as 15. It is even worse for general $p \in (1,\infty)$ if we try to enumerate the vertices of $X$ for finding the maximal p-norm, for the number of vertices can be as much as $\binom{m}{n}$ which, by Stirling's formula, is bounded below by an exponential in $n$ for $m \geqq (1+\varepsilon)n$ for any constant positive $\varepsilon$. One may try to find other algorithms that are computationally effective. Unfortunately, as shown in the next section, problem (1.3) with $p \neq \infty$ is no easier than the partition problem (see (4.1) below) which is inherently intractable.

## 4. The intractability of the norm maximization problem for $p \neq \infty$

We begin this section with some basic concepts of complexity theory [6, 11]. Problem A reduces (in polynomial time) to problem B, denoted by $A \propto B$, iff the following holds: If there is a polynomial time algorithm for B, then one can construct a polynomial time algorithm for A using the algorithm for B as a subroutine. Problems A and B are polynomially equivalent iff $A \propto B$ and $B \propto A$. An NP-complete problem is one which is polynomially equivalent to any one of the standard intractable problems such as the satisfiability, partition, or travelling salesman problems [6, 11]. These problems are considered intractable because any known algorithm which solves any one of them requires, in the worst case, an amount of time which is not bounded above by any polynomial in problem size. An NP-hard problem is any problem such that all problems in NP reduce to it in polynomial time. For details see [6, Chapter 5]. Thus an NP-hard problem is at least as difficult as an NP-complete problem. We will now show that our norm maximization problem (1.3) is NP-hard for $p \neq \infty$ by reducing the following NP-complete partition problem to it:

(4.1) Given integers $c_1, c_2, \ldots, c_n$, is there a set $S \subset \{1,2,\ldots,n\}$ such that $\sum_{j \in S} c_j = \sum_{j \notin S} c_j$ ?

**4.1 Theorem.** The norm maximization problem (1.3) is NP-hard for $p \in [1,\infty)$.

*Proof.* We will show this by reducing (4.1) to (1.3). Let $p \in [1,\infty)$. We first reduce (4.1) to the following problem:

(4.2) Given integers $c_1, c_2, \ldots, c_n$, is there an $x \in R^n$ such that:

$$\sum_{i=1}^{n} c_i x_i = 0, \ -1 \leqq x_i \leqq 1, \ 1 \leqq i \leqq n, \ \|x\|_p^p \geqq n \ ?$$

It is easy to see that (4.1) has a solution $S$ iff (4.2) has a solution $x$ with $|x_i| = 1$

for $1 \leq i \leq n$ and $x_i = 1$ for $i \in S$ and $x_i = -1$ for $i \notin S$. Now it is easy to see that (4.2) can be reduced to an instance of problem (1.3) by defining

$$A := \begin{bmatrix} I \\ -I \\ c^T \\ -c^T \end{bmatrix}, \quad b := \begin{bmatrix} -e \\ -e \\ 0 \\ 0 \end{bmatrix}$$

and answering the question:

(4.3) $$\text{Is } \max\{\|x\|_p^p \mid x \in R^n, Ax \geq b\} \geq n \quad ?$$

Hence if we can solve (1.3) in polynomial time we can solve each of (4.3), (4.2) and (4.1) in polynomial time. Hence (4.1) $\propto$ (1.3) and (1.3) is NP-hard. □

We go on to show now that our norm maximization problem (1.3) is in fact NP-complete for integer $p \neq \infty$. In order to do this we introduce additional concepts from complexity theory. A <u>nondeterministic</u> algorithm is an algorithm which at each step has a finite number of moves from which to choose (instead of only one for deterministic algorithms) and it solves a problem in a finite sequence of choices leading to a correct answer. <u>NP</u> is the class of problems solvable by a nondeterministic algorithm in polynomial time, including (4.1) and all other NP-complete problems. In fact NP-complete problems are the class of most difficult problems in NP in the sense that each problem in NP reduces in polynomial time to each NP-complete problem. By Cook's theorem [1, 6, 11], all we need to show for (1.3) to be NP-complete is that it is NP-hard (which we already have done in Theorem 4.1) and that it is in the class NP, which we proceed to do now. In order to do that we introduce the following <u>decision</u> problem related to our optimization problem (1.3):

(4.4) Given A, b with integer entries satisfying (1.2), and nonzero integers r, s, p, is there a vector x in $R^n$ such that

$$Ax \geq b, \ \|x\|_p^p \geq \frac{r}{s} \quad ?$$

Note that in the proof of Theorem 4.1 we have already established that the decision problem (4.4) is NP-hard, because we reduced the partition problem (4.1) to (4.2) which is an instance of (4.4). We will now first show that (4.4) is in NP and hence it is NP-complete. Then we will show that an optimization problem (1.3) is polynomially equivalent to the NP-complete decision problem (4.4). Note that condition (1.2) which is imposed on problem (4.4) which is a necessary and sufficient condition for the boundedness of $X$, plays an essential role in Proposition (4.2) below which establishes that (4.4) is in NP.

4.2 *Proposition.* Problem (4.4) is in NP for integer $p \geq 1$.

*Proof.* It follows by the convexity of the norm and the boundedness of $X$ by (1.2) [13], that $\|x\|_p^p \geq \frac{r}{s}$ for some $x \in X$ iff $\|v\|_p^p \geq \frac{r}{s}$ for some vertex $v$ of $X$. Moreover, $v$ is a vertex iff there is a $J$ $\{1,2,\ldots,m\}$, $|J| = n$ such that $v$ is the unique solution of $A_i x = b_i$, $i \in J$, and $A_j x \geq b_j$ for $j \notin J$. Consequently we can prescribe the following nondeterministic algorithm for solving (4.4).

4.3 *Algorithm*

(i) *choose* $J$, a subset of $\{1,2,\ldots,m\}$ with cardinality $n$.

(ii) Solve $A_i x = b_i$, $i \in J$ for one $x$, or conclude that the system is inconsistent.

(iii) *if* solution $x$ found *and* $A_j x \geq b_j$ for $j \notin J$ *and* $\|x\|_p^p \geq \frac{r}{s}$ then *print* $x$; *success*; *else* *failure*; *endif*.

Step (ii) can be performed in polynomial time (e.g. by Gaussian elimination). Since we have assumed that $p$ is an integer, $\|x\|_p^p$ can be evaluated in polynomial time. Hence Algorithm 4.3 is a polynomial time algorithm and (4.4) is in NP.

□

In standard terminology, the terms NP and NP-complete refer to decision problems only but not to optimization problems. Now we show that the NP-complete decision problem (4.4) and our optimization problem (1.3) are polynomially equivalent. First it is obvious that if one can solve the optimization problem (1.3), then one can answer the decision problem (4.4). The reverse is usually done by a binary search technique showing that the optimization problem can be solved by a polynomial number of decision problems. This is all rather obvious for discrete combinatorial problems, but not for our continuous problem

(1.3). To do this here, we shall use arguments similar to those of Khachian [7]. Define

$$L := \sum_{i,j=1}^{m,n} \log_2(|A_{ij}| + 1) + \sum_i \log_2(|b_i| + 1) + \log_2(nm+1) + \log_2(p+1) .$$

$L$ is the total length of binary digits representing the input $A, b, n, m, p$ of problem (1.3).

4.4 Theorem. For any integer $p \geq 1$, problem (1.3) is polynomially equivalent to the NP-complete decision problem (4.4).

Proof. Since an optimal solution of (1.3) is at a vertex of $X$ [13], such a vertex can be written by Cramer's rule as $\left(\frac{D_1}{D}, \frac{D_2}{D}, \ldots, \frac{D_n}{D}\right)^T$, where $D$ and $D_i$ are determinants of submatrices of $[A \quad b]$. Hence

(i) For any vertex $v = \left(\frac{D_1}{D}, \ldots, \frac{D_n}{D}\right)^T$, $|D| < 2^L$, $|D_i| < 2^L$, $\|v\|_p^p < 2^{pL}$. (See [5] for details.)

(ii) For any two distinct vertices $\|v\|_p \neq \|w\|_p$, $v = \left(\frac{D_1}{D}, \ldots, \frac{D_n}{D}\right)^T$, $w = \left(\frac{B_1}{B}, \ldots, \frac{B_n}{B}\right)^T$ it follows that

$$\left|\|v\|_p^p - \|w\|_p^p\right| = \left|\frac{|D_1|^p + \ldots + |D_n|^p}{|D|^p} - \frac{|B_1|^p + \ldots + |B_n|^p}{|B|^p}\right| \geq \frac{1}{|D|^p|B|^p} > 2^{-2pL}$$

Hence we can reduce (1.3) to (4.4) by binary search on the interval $[0, 2^{pL}]$ until the range is less than $2^{-2pL}$. Since each iteration reduces range by half, $3pL$ iterations will do that by the following:

4.5 Algorithm

(i) $\ell \leftarrow 0$, $u \leftarrow 2^{pL}$

(ii) for $i = 1$ to $3pL$ do

(iii) solve the decision problem (4.4) for input $A$, $b$, $\frac{r}{s} = \frac{1}{2}(\ell+u)$

(iv) if answer is yes then $\ell \leftarrow \frac{r}{s}$ else $u \leftarrow \frac{r}{s}$ endif

(v) end for

If (iii) can be done in polynomial time, then (i) to (v) can be done in polynomial time. After (v), we known that there exists an $x \in X$ such that $\ell = u-2^{-2pL}$, $\|x\|_p^p \geq \ell$, whereas there is no $x \in X$ such that $\|x\|_p^p \geq u$. Hence if we now use Algorithm 4.3 with input $\frac{r}{s} = \ell$, A and b, the x printed in step (iii) of Algorithm 4.3 is an exact vertex solution of (1.3) obtained in polynomial time. Hence (1.3) reduces to (4.4).

□

REFERENCES


1. S. A. Cook: "The complexity of theorem proving procedures", Proceedings 3rd ACM Symposium on the Theory of Computing ACM. 1971, 151-158.

2. R. W. Cottle and G. B. Dantzig: "Complementary pivot theory of mathematical programming", Linear Algebra and Its Applications 1, 1968. 103-125.

3. S. J. Chung and K. G. Murty: "Polynomially bounded ellipsoid algorithms for convex quadratic programming" in O. L. Mangasarian, R. R. Meyer and S. M. Robinson (editors) "Nonlinear programming 4", Academic Press 1981, 439-485.

4. G. B. Dantzig: "Linear programming and extensions". Princeton University, Princeton, New Jersey 1963.

5. P. Gâcs and L. Lovâsz: "Khachian's algorithm for linear programming", Mathematical Programming Study 14, 1981, 61-68.

6. M. R. Garey and D. S. Johnson: "Computers and intractability: a guide to the theory of NP-completeness", W. H. Freeman, San Francisco, California 1979.

7. L. G. Khachian: "A polynomial algorithm in linear programming", Doklady Akademiia Nauk SSR 244:S, 1979. 1093-1093. Translated in Soviet Mathematics Doklady 20 (1979) 191-194.

8. O. L. Mangasarian: "Characterizations of bounded solutions of linear complementarity problems", Mathematical Programming Study 19, 1982. 153-166.

9. O. L. Mangasarian: "Simple computable bounds for solutions of linear complementarity problems and linear programs", University of Wisconsin Computer Sciences Tech. Report 519, October 1983, to appear in Mathematical Programming Study.

10. K. G. Murty: "Linear programming", Wiley, New York, 1983.

11. C. H. Papadimitriou and K. Steiglitz: "Combinatorial optimization: algorithms and complexity", Prentice-Hall, Englewood Cliffs, New Jersey 1982.

12. S. M. Robinson: "A characterization of stability in linear programming", Operations Research 25, 1977, 435-447.

13. R. T. Rockafellar: "Convex analysis", Princeton University Press, Princeton, New Jersey 1970.

14. A. C. Williams: "Marginal values in linear programming", Journal SIAM 11, 1963, 82-94.

15. A. C. Williams: "Boundedness relations for linear constraint sets", Linear Algebra and Its Applications 3, 1970, 129-141.

16. A. C. Williams: "Complementarity theorems for linear programming", SIAM Review 12, 1970, 135-137.

OLM/THS/jvs