Mathematical Programming 39 (1987) 117-129
North-Holland

SOME NP-COMPLETE PROBLEMS IN QUADRATIC AND NONLINEAR PROGRAMMING

Katta G. MURTY*

Department of Industrial and Operations Engineering, The University of Michigan, 1205 Beal Avenue, Ann Arbor, MI 48109-2117, USA

Santosh N. KABADI**

Faculty of Administration, University of New Brunswick, Fredericton, NB, Canada E3B 5A6

Received 13 December 1985
Revised manuscript received 9 March 1987

In continuous variable, smooth, nonconvex nonlinear programming, we analyze the complexity of checking whether

(a) a given feasible solution is not a local minimum, and

(b) the objective function is not bounded below on the set of feasible solutions.

We construct a special class of indefinite quadratic programs, with simple constraints and integer data, and show that checking (a) or (b) on this class is NP-complete. As a corollary, we show that checking whether a given integer square matrix is not copositive, is NP-complete.

Key words: Nonconvex nonlinear programming, local minimum, global minimum, copositive matrices, NP-complete.

1. Introduction

Consider the smooth nonlinear program (NLP)

$$\begin{aligned}&\text{minimize} \quad \theta(x)\\&\text{subject to} \quad g_i(x) \geqslant 0, \ i=1 \text{ to } m\end{aligned} \tag{1}$$

where each of the functions is a real valued function defined on $\mathbb{R}^n$, with high degree of differentiability. This NLP is called

a *convex NLP*, if $\theta(x)$ is convex, and $g_i(x)$ is concave for all i,

a *nonconvex NLP*, otherwise.

Under some constraint qualifications, necessary and sufficient optimality conditions are known for convex NLPs [2, 4, 7]. Using them, it is possible to check efficiently whether a given feasible solution satisfying the constraint qualifications is a (global) optimum solution or not.

* Research partially supported by NSF Grants No. ECS-8401081 and ECS-8521183.
** Research partially supported by NSERC (Canada) Grant No. A8085.

We will use the following abbreviations in this paper.

NLP -Nonlinear Program
CQ -Constraint Qualifications
PSD -Positive Semidefinite
QP -Quadratic Program
LCP -Linear Complementarity Problem
BFS -Basic Feasible Solution

For nonconvex NLPs, under some CQ, necessary conditions for a local minimum are known, and there are some sufficient conditions [2, 4, 7]. There are no known simple conditions, which are both necessary and sufficient for a given point to be a local minimum. The complexity of checking whether a given feasible solution is a local minimum is not usually addressed in the literature. In fact, many text books in nonlinear programming leave the reader with the impression that algorithms converge to a global minimum in convex NLPs, and to a local minimum in nonconvex NLPs. The documentation for many professional NLP software packages also creates the same impression, which could be quite erroneous.

In this paper, we examine the computational complexity of determining whether a given feasible solution is not a local minimum, and that of determining whether the objective function is not bounded below on the set of feasible solutions, in this class of problems. For this purpose, we analyze an indefinite QP with integer data, which may be considered as the simplest nonconvex NLP. On this problem, the above questions can be studied using the discrete techniques of computational complexity theory, and in fact we will show that these questions are NP-complete. This clearly shows that on a general smooth nonconvex NLP, the questions mentioned above are "hard problems", as defined in computational complexity theory. Thus, in nonconvex minimization, even the down-to-earth goal of guaranteeing that a local minimum will be obtained by an algorithm (as opposed to the lofty goal of finding the global minimum) may be hard to attain. We make some more comments on this issue at the end of the paper.

2. Finding a global minimum in a smooth nonconvex NLP is a hard problem

Computing a global minimum, or checking whether a given feasible solution is a global minimum, for a smooth nonconvex NLP, may be hard problems in general. We provide two examples. Example 1 is an interesting digression. It refers to a famous unsolved problem in mathematics, but one which has not been formally shown to be a "hard problem" in the usual complexity sense. In Example 2 we formally establish that finding a global minimum in a nonconvex NLP is a hard problem, by showing that a well known NP-complete problem can be posed as a special case of it.

Example 1. *Fermat's Last Theorem.* Some of the most difficult unsolved problems in mathematics can be posed as problems of finding a global minimum in a smooth nonconvex NLP. Consider Fermat's last theorem, unresolved since the year 1637. It states that there exists no positive integer solution (x, y, z) to the equation

$$x^n + y^n = z^n$$

when n is an integer $\geqslant 3$ (here, $x, y, z \in \mathbb{R}^1$). Even though this conjecture has been shown to be true for several individual values of n, in general, it remains open. Obviously, Fermat's last theorem is not true iff the global minimum objective value in the following NLP is 0 and attained where α is a positive penalty parameter.

$$\begin{aligned} \text{minimize} \quad & (x^n + y^n - z^n)^2 \\ & + \alpha((1-\cos(2\pi x))^2 + (1-\cos(2\pi y))^2 + (1-\cos(2\pi z))^2 \\ & + (1-\cos(2\pi n))^2) \\ \text{subject to} \quad & x, y, z \geqslant 1, \ \ n \geqslant 3. \end{aligned}$$

This problem uses transcendentals, and therefore involves a different model of computation. Even though this example transforms a famous unsolved problem in mathematics into a nonconvex NLP, it does not mathematically establish that computing a global minimum is a hard problem, since Fermat's last theorem is not known to be complete for any class.

Example 2. *Subset Sum Problem.* This is a problem in discrete optimization which is known to be NP-complete [3]: given positive integers $d_0; d_1, \ldots, d_n$; is there a solution to

$$\begin{aligned} & \sum_{i=1}^{n} d_j y_j = d_0, \\ & y_j = 0 \text{ or } 1 \text{ for all } j. \end{aligned} \tag{2}$$

Now consider the following QP:

$$\begin{aligned} \text{minimize} \quad & \left(\sum_{j=1}^{n} d_j y_j - d_0 \right)^2 + \sum_{j=1}^{n} y_j(1-y_j) \\ \text{subject to} \quad & 0 \leqslant y_j \leqslant 1, \ j = 1 \text{ to } n. \end{aligned} \tag{3}$$

Because of the second term in the objective function, (3) is a nonconvex QP. Clearly (2) has a feasible solution iff the global minimum objective value in (3) is zero. Checking whether (2) has a feasible solution is NP-complete, and hence, computing the global minimum in (3), a very special case of a smooth nonconvex NLP, is an NP-hard problem. This formally establishes that in general, computing a global minimum in a smooth nonconvex NLP is a hard problem.

3. Can we compute efficiently a local minimum for a smooth nonconvex NLP?

We will now study the question of whether it is possible to efficiently

(a) compute a local minimum, or

(b) check whether a given feasible solution for such a problem is not a local minimum.

To do this, we first review the known optimality conditions for a given feasible solution $\bar{x}$ of (1) to be a local minimum. Let $A = \{i\colon g_i(\bar{x}) = 0\}$. Optimality conditions are derived under the assumption that some CQ [2, 4, 7] are satisfied at $\bar{x}$, which we assume.

First order necessary conditions for $\bar{x}$ to be a local minimum for (1)

There must exist a $\bar{\mu}_A = (\bar{\mu}_i\colon i \in A)$ such that

$$\begin{aligned} &\nabla\theta(\bar{x}) - \sum_{i\in A} \bar{\mu}_i \nabla g_i(\bar{x}) = 0, \\ &\bar{\mu}_i \geqslant 0 \quad \text{for all } i \in A. \end{aligned} \tag{4}$$

Here $\nabla\theta(\bar{x})$, $\nabla g_i(\bar{x})$ are the gradient vectors (row vectors) of these functions evaluated at $\bar{x}$. Given the feasible solution $\bar{x}$, it is possible to check efficiently whether (4) holds, using any of the available polynomial time algorithms for linear programming. A feasible solution $\bar{x}$ is called a *KKT point* for (1) if these first order necessary conditions hold at $\bar{x}$.

Second order necessary conditions for $\bar{x}$ to be a local minimum for (1)

These conditions include (4). Given $\bar{\mu}_A$ satisfying (4) together with $\bar{x}$, let

$$L(x, \bar{\mu}_A) = \theta(x) - \sum_{i\in A} \bar{\mu}_i g_i(x).$$

In addition to (4), in these conditions we require

$$y^{\mathrm{T}}Hy \geqslant 0 \quad \text{for all } y \in \{y\colon \nabla g_i(\bar{x})y = 0 \text{ for each } i \in A\}, \tag{5}$$

where H is the Hessian matrix (the matrix of second partial derivatives) of $L(x, \bar{\mu}_A)$ with respect to x at $x = \bar{x}$. Condition (5) requires the solution of a quadratic program involving only equality constraints. It is equivalent to checking the positive semi-definiteness of a matrix which can be carried out efficiently [7, 8, 9].

Sufficient conditions for $\bar{x}$ to be a local minimum for (1)

Given the feasible solution $\bar{x}$ for (1), and the multiplier vector $\bar{\mu}_A$, which together satisfy (4) and (5), define H as above and let T^1 be the set of feasible solutions of

$$\begin{aligned} &y \neq 0, \\ &\nabla g_i(\bar{x})y = 0 \quad \text{for each } i \in A \text{ such that } \bar{\mu}_i > 0, \\ &\nabla g_i(\bar{x}) \geqslant 0 \quad \text{for each } i \in A \text{ such that } \bar{\mu}_i = 0. \end{aligned}$$

The most general sufficient optimality condition known states that if

$$y^{\mathrm{T}}Hy > 0 \quad \text{for all } y \in \boldsymbol{T}^1 \tag{6}$$

then $\bar{x}$ is a local minimum for (1). Unfortunately when H is not PSD, checking whether (6) holds is equivalent to a nonconvex QP which may be hard to solve, as we will see later (Theorem 4).

Aside from the question of the difficulty of checking whether (6) holds, we can verify that the gap between conditions (5) and (6) is very wide, particularly when the set $\{i: i \in \boldsymbol{A} \text{ and } \bar{\mu}_i = 0\} \neq \emptyset$. In this case, condition (5) may hold, and even if we are able to check (6), if it is not satisfied, present theory does not enable us to determine whether $\bar{x}$ is a local minimum for (1).

The questions investigated

We will now study the following questions:

(i) Given a smooth nonconvex NLP, can we check efficiently whether a given feasible solution is a local minimum or not?

(ii) At least in the simple case when the constraints are linear, can we check efficiently whether the objective function is bounded below or not on the set of feasible solutions?

We will use a simple indefinite QP for our investigation. Let D be an integer square matrix of order n. D is PSD iff $x^{\mathrm{T}}Dx \geqslant 0$ for all $x \in \mathbb{R}^n$. So, checking whether D is PSD involves the decision problem

$$\text{is there an } x \in \mathbb{R}^n \text{ satisfying } x^{\mathrm{T}}Dx < 0?$$

It is well known that this question can be settled by performing at most n Gaussian pivot steps along the main diagonal of D, requiring a computational effort of at most $O(n^3)$ [7 or 8].

The matrix D is said to be copositive if $x^{\mathrm{T}}Dx \geqslant 0$ for all $x \geqslant 0$. All PSD matrices are copositive, but the converse may not be true. Testing whether the given matrix D is not copositive involves the decision problem

$$\text{is there an } x \geqslant 0 \text{ satisfying } x^{\mathrm{T}}Dx < 0?$$

If D is not PSD, no efficient algorithm is known for this problem (some enumerative methods are available [7], but the computational effort required by these methods grows exponentially with n in the worst case). In fact we show later that this decision problem is NP-complete. To study this decision problem, we are naturally led to the following QP.

$$\begin{aligned} &\text{minimize} \quad Q(x) = x^{\mathrm{T}}Dx \\ &\text{subject to} \quad x \geqslant 0. \end{aligned} \tag{7}$$

We will show that this QP is an NP-hard problem.

We assume that D is not PSD, so $Q(x)$ is nonconvex and (7) is a nonconvex QP, in fact is can be considered the *simplest nonconvex NLP.* We consider the following decision problems.

Problem 1. Is $x=0$ not a local minimum for (7)?

Problem 2. Is $Q(x)$ not bounded below on the set of feasible solutions of (7)?

Clearly, the answer to Problem 2 is in the affirmative iff the answer to Problem 1 is. We will show that both these problems are NP-complete. To study Problem 1, we can replace (7) by the QP.

$$\begin{aligned} &\text{minimize} \quad Q(x)=x^{\mathrm{T}}Dx \\ &\text{subject to} \quad 0\leqslant x_j\leqslant 1, \ j=1 \text{ to } n. \end{aligned} \tag{8}$$

Lemma 1. *The decision problem "is there an x feasible to* (8) *which satisfies* $Q(x)<0$", *is in the class* NP.

Proof. Given an x feasible to (8), checking whether $Q(x)<0$ can be done by computing $Q(x)$ which takes $O(n^2)$ time. If the answer to the decision problem is in the affirmative, an optimum solution x of (8) satisfies $Q(x)<0$. There is an LCP corresponding to (8), and an optimum solution for (8) must correspond to a BFS for this LCP. There are only a finite number of BFSs for this LCP, and they are all rational vectors of polynomial length relative to the input size of (8). So, a nondeterministic algorithm can find one of them satisfying $Q(x)<0$ (if such a BFS exists), in polynomial time. Hence this problem is in the class NP. □

Lemma 2. *The optimum objective value in* (8) *is either* 0 *or* $\leqslant -2^{-L}$ *where* L *is the size of* D.

Proof. Since the set of feasible solutions of (8) is a compact set and $Q(x)$ is continuous, (8) has an optimum solution.

By well known results, the necessary optimality conditions for (8) lead to the following LCP [7 or 8].

$$\begin{pmatrix} u \\ \cdots \\ v \end{pmatrix} - \begin{pmatrix} D & \vdots & I \\ \cdots & & \cdots \\ -I & \vdots & 0 \end{pmatrix} \begin{pmatrix} x \\ \cdots \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ \cdots \\ e \end{pmatrix} \tag{9}$$

$$\begin{pmatrix} u \\ \cdots \\ v \end{pmatrix} \geqslant 0, \quad \begin{pmatrix} x \\ \cdots \\ y \end{pmatrix} \geqslant 0, \tag{10}$$

$$\begin{pmatrix} u \\ \cdots \\ v \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} x \\ \cdots \\ y \end{pmatrix} = 0, \tag{11}$$

where y is the column vector of Lagrange multipliers associated with the constraints "$x_j \leqslant 1$ for all j"; u, v, are the column vectors in $\mathbb{R}^n$ of dual and primal slack variables; and e is the column vector of all 1's in $\mathbb{R}^n$. For every optimum solution x of (8), there exist vectors u, v, y such that (u, v, x, y) solves (9), (10) and (11). Also, it can be verified that whenever (u, v, x, y) satisfies (9), (10) and (11), $x^{\mathrm{T}}Dx = -e^{\mathrm{T}}y$. Thus, there exists an optimum solution of (8) which is a BFS of (9), (10). By the results under the ellipsoid algorithm [6, 7], in every BFS of (9), (10), each y_j is either 0 or $\geqslant 2^{-L}$. If the optimum objective value in (8) is not zero, it must be <0, and this together with the above facts implies that an optimum solution x of (8) corresponds to a BFS (u, v, x, y) of (9), (10) in which $-e^{\mathrm{T}}y < 0$. All these facts clearly imply that the optimum objective value in (8) is either 0 or $\leqslant -2^{-L}$. □

We now make a list of several decision problems, some of which we have already seen, and some new ones which we need to establish our results. Problem 5 is the subset sum problem with data d_0; $d_1, \ldots, d_n$ (all positive integers) defined earlier.

Problem 3. Is there an $x \geqslant 0$ satisfying $Q(x) < 0$?

Problem 4. For any positive integer a_0, is there an $x \in \mathbb{R}^n$ satisfying $e^{\mathrm{T}}x = a_0$, $x \geqslant 0$ and $Q(x) < 0$?

Problem 5. *Subset sum problem.* Is there an integer vector $y = (y_j) \in \mathbb{R}^n$ satisfying

$$\sum_{j=1}^{n} d_j y_j = d_o, \quad 0 \leqslant y_j \leqslant 1, \; j = 1 \text{ to } n.$$

Let δ be a positive integer satisfying

$$\delta > 4\left(d_o\left(\sum_{j=1}^{n} d_j\right)\right)^2 n^3.$$

Let l be the size of this subset sum problem, that is, the total number of digits in all the data for the problem. Let ε be a positive rational number $<2^{-nl^2}$.

Now we define several functions that will be needed in the proofs to follow. They involve nonnegative variables $y = (y_1, \ldots, y_n)^{\mathrm{T}}$ and $s = (s_1, \ldots, s_n)^{\mathrm{T}}$, related to the subset sum problem.

$$\begin{aligned} f_1(y, s) &= \left(\sum_{j=1}^{n} d_j y_j - d_o\right)^2 + \delta\left(\sum_{j=1}^{n} (y_j + s_j - 1)^2\right) + \sum_{j=1}^{n} y_j s_j \\ &= \left(\sum_{j=1}^{n} d_j y_j\right)^2 + \sum_{j=1}^{n} y_j s_j + \delta \sum_{j=1}^{n} (y_j + s_j)^2 - 2d_o\left(\sum_{j=1}^{n} d_j y_j\right) \\ &\quad - 2\delta \sum_{j=1}^{n} (y_j + s_j) + n\delta + d_o^2, \end{aligned}$$

$$\begin{aligned} f_2(y, s) &= f_1(y, s) + 2d_o\left(\sum_{j=1}^{n} d_j y_j (1 - y_j)\right) \\ &= \left(\sum_{j=1}^{n} d_j y_j\right)^2 + \delta \sum_{j=1}^{n} (y_j + s_j)^2 + \sum_{j=1}^{n} y_j s_j - 2d_o\left(\sum_{j=1}^{n} d_j y_j^2\right) \end{aligned}$$

$$-2\delta \sum_{j=1}^{n} (y_j+s_j)+n\delta+d_o^2,$$

$$f_3(y,s)=\left(\sum_{j=1}^{n} d_j y_j\right)^2+\delta \sum_{j=1}^{n} (y_j+s_j)^2+\sum_{j=1}^{n} y_j s_j-2d_o \sum_{j=1}^{n} d_j y_j^2+d_o^2-n\delta,$$

$$f_4(y,s)=\left(\sum_{j=1}^{n} d_j y_j\right)^2+\delta \sum_{j=1}^{n} (y_j+s_j)^2+\sum_{j=1}^{n} y_j s_j-2d_o \sum_{j=1}^{n} d_j y_j^2$$
$$+\left(\frac{d_o^2-n\delta}{n^2}\right)\left(\sum_{j=1}^{n} (y_j+s_j)\right)^2,$$

$$f_5(y,s)=f_4(y,s)-\left(\frac{\varepsilon}{n^2}\right)\left(\sum_{j=1}^{n} (y_j+s_j)\right)^2.$$

Let

$$\boldsymbol{P}=\left\{(y,s)\colon y=(y_j)\geq 0,\ s=(s_j)\geq 0,\ \sum_{j=1}^{n} (y_j+s_j)=n\right\}.$$

These allow stating the following additional decision problems.

Problem 6. Is there an $(y,s)\in \boldsymbol{P}$ satisfying $f_1(y,s)\leq 0$?

Problem 7. Is there an $(y,s)\in \boldsymbol{P}$ satisfying $f_2(y,s)\leq 0$?

Problem 8. Is there an $(y,s)\in \boldsymbol{P}$ satisfying $f_4(y,s)\leq 0$?

Problem 9. Is there an $(y,s)\in \boldsymbol{P}$ satisfying $f_5(y,s)<0$?

Here is a summary of what we will prove next. In Theorem 1 we show that Problem 4 is NP-hard. In Theorem 2 we combine the results from Theorem 1 and Lemma 1 and show that Problems 1, 2, 3, 4 are all NP-complete. In Theorem 3 we establish that checking whether an integer square matrix D is not copositive is NP-complete. Phrasing this in the affirmative, this shows that checking whether an integer square matrix D is copositive, is in the co-NP-complete class.

Theorem 1. *Problem* 4 *is an* NP-*hard problem.*

Proof. Since $f_1(y,s)$ is a sum of nonnegative terms whenever $(y,s)\in \boldsymbol{P}$, if $(\bar{y},\bar{s})\in \boldsymbol{P}$ satisfies $f_1(\bar{y},\bar{s})\leq 0$, then we must have $f_1(\bar{y},\bar{s})=0$. From the definition of $f_1(y,s)$, this clearly implies that the following conditions must hold.

$$\sum_{j=1}^{n} d_j\bar{y}_j=d_o,\quad \bar{y}_j\bar{s}_j=0\quad\text{and}\quad \bar{y}_j+\bar{s}_j=1\quad\text{for all } j=1 \text{ to } n.$$

These conditions clearly imply that $\bar{y}$ is a solution of the subset sum problem and that the answer to Problem 5 is in the affirmative. Conversely if $\hat{y}=(\hat{y}_j)$ is a solution to the subset sum problem, define $\hat{s}=(\hat{s}_j)$ where $\hat{s}_j=1-\hat{y}_j$ for each $j=1$ to n, and it can be verified that $f_1(\hat{y},\hat{s})=0$. This verifies that Problems 5 and 6 are equivalent.

Whenever $\bar{y}$ is a 0–1 vector, we have $\bar{y}_j = \bar{y}_j^2$ for all j, and this implies that $f_1(\bar{y}, s) = f_2(\bar{y}, s)$ for any s. So, from the above arguments, we see that if $(\bar{y}, \bar{s}) \in \boldsymbol{P}$ satisfies $f_1(\bar{y}, \bar{s}) \leqslant 0$, then $f_1(\bar{y}, \bar{s}) = f_2(\bar{y}, \bar{s}) = 0$. If $0 \leqslant y_j \leqslant 1$, we have $2d_o d_j y_j(1-y_j) \geqslant 0$. If $(y, s) \in \boldsymbol{P}$, and $y_j > 1$, then $(\delta/2)(y_j + s_j - 1)^2 + 2d_o d_j y_j(1-y_j) \geqslant 0$, since δ is large. Using this and the definitions of $f_1(y, s)$, $f_2(y, s)$, it can be verified that for $(y, s) \in \boldsymbol{P}$, if $f_2(y, s) \leqslant 0$ then $f_1(y, s) \leqslant 0$ too. These facts imply that Problems 6 and 7 are equivalent.

Clearly, Problems 7 and 8 are equivalent. From the definition of ε and using Lemma 2, one can verify that Problems 8 and 9 are equivalent.

Problem 9 is a special case of Problem 4. Since Problem 5 is NP-complete, from the above chain of arguments we conclude that Problem 4 is NP-hard. □

Theorem 2. *Problems* 1, 2, 3, 4 *stated above are all* NP-*complete problems.*

Proof. The answer to Problem 4 is in the affirmative, iff the answer to the decision problem in the statement of Lemma 1 is in the affirmative. From Lemma 1 we conclude that Problem 4 is in NP. From Theorem 1, this shows that Problem 4 is NP-complete.

Problems 3 and 4 are clearly equivalent, so Problem 3 is NP-complete too. Problems 1, 2 are both equivalent to Problem 3, so Problems 1, 2 are also NP-complete. □

Theorem 3. *Given an integer square matrix D, the decision problem "is D not copositive"? is* NP-*complete.*

Proof. The decision problem "is D not copositive"? is equivalent to Problem 1, hence this result follows from Theorem 2. □

Theorem 4. *Let $\bar{x}$ be a given feasible solution of* (1), *and $\boldsymbol{A} = \{i\colon g_i(\bar{x}) = 0\}$. If H, and $\nabla g_i(\bar{x})$ for each $i \in \boldsymbol{A}$, are rational, checking whether the sufficient optimality condition* (6) *holds, is co-*NP-*complete.*

Proof. Let P, B, E be given rational matrices of orders $n \times n$, $m \times n$, $p \times n$ respectively. Consider the following decision problem.

Problem 10. Is there an $x \in \mathbb{R}^n$ satisfying $Bx = 0$, $Ex \geqslant 0$, $x^{\mathrm{T}}Px < 0$?

Using arguments similar to those in Lemmas 1, 2, it can be shown that Problem 10 is in the class NP. Also, Problem 4 is a special case of Problem 10. So, by Theorem 2, Problem 10 is NP-complete. Under the hypothesis of the theorem, checking whether (6) holds is the complementary problem of Problem 10, and is therefore co-NP-complete. □

4. Can we efficiently check local minimality in unconstrained minimization problems?

In Section 3 we discussed constrained optimization problems. In this section, we will show that results corresponding to those proved in Section 3, hold even for unconstrained optimization problems.

Let $\theta(x)$ be a real valued smooth function defined on $\mathbb{R}^n$. Let $H(\theta(x))$ denote the Hessian matrix of $\theta(x)$ at x. Consider the unconstrained NLP

$$\text{minimize} \quad \theta(x). \tag{12}$$

A necessary condition for a given point $\bar{x} \in \mathbb{R}^n$ to be a local minimum for (12) is

$$\nabla\theta(\bar{x}) = 0 \quad \text{and} \quad H(\theta(\bar{x})) \text{ is PSD.} \tag{13}$$

A sufficient condition for $\bar{x}$ to be a local minimum for (12) is

$$\nabla\theta(\bar{x}) = 0 \quad \text{and} \quad (H(\theta(\bar{x})) \text{ is positive definite.} \tag{14}$$

Both conditions (13) and (14) can be checked very efficiently. If (13) is satisfied, but (14) is violated, there are no known simple conditions to check whether or not $\bar{x}$ is a local minimum for (12). Here, we investigate the complexity of checking whether or not a given point $\bar{x}$ is a local minimum for (12), and that of checking whether $\theta(x)$ is bounded below over $\mathbb{R}^n$.

As before, let $D = (d_{ij})$ be an integer square symmetric matrix of order n. Consider the unconstrained problem,

$$\text{minimize} \quad h(u) = (u_1^2, \ldots, u_n^2) D (u_1^2, \ldots, u_n^2)^{\mathrm{T}}. \tag{15}$$

Clearly, (15) is an instance of the general unconstrained minimization problem (12). Consider the following decision problems.

Problem 11. Is $\bar{u} = 0$ not a local minimum for (15)?

Problem 12. Is $h(u)$ not bounded below on $\mathbb{R}^n$?

We have, for $i, j = 1$ to n,

$$\frac{\partial h(u)}{\partial u_j} = 4u_j((u_1^2, \ldots, u_n^2)D_{.j}),$$

$$\frac{\partial^2 h(u)}{\partial u_i\, \partial u_j} = 8u_i u_j d_{ij}, \quad i \neq j$$

$$\frac{\partial^2 h(u)}{\partial u_j^2} = 4(u_1^2, \ldots, u_n^2)D_{.j} + 8u_j^2 d_{jj},$$

where $D_{.j}$ is the jth column vector of D. So, $\bar{u} = 0$ satisfies the necessary conditions for being a local minimum for (15), but not the sufficient condition given in (14).

Using the transformation $x_j = u_j^2$, $j = 1$ to n, we see that (15) is equivalent to (7). So Problems 1 and 11 are equivalent. Likewise, Problems 2 and 12 are equivalent. By Theorem 2, we conclude that both Problems 11 and 12 are NP-hard. Thus, even in unconstrained minimization, to check whether the objective function is not bounded below, and to check whether a given point is not a local minimum, may be hard problems in general. This also shows that checking whether a given smooth nonlinear function (even a polynomial) is or is not locally convex at a given point, may be a hard problem in general.

5. What are suitable goals for algorithms in nonconvex NLP?

Much of the nonlinear programming literature stresses that the goal for algorithms for solving nonconvex NLPs should be to obtain a local minimum. Our results here show that in general, this may be hard to guarantee.

Many nonlinear programming algorithms are iterative in nature, that is, beginning with an initial point x^0, they obtain a sequence of points $\{x^r: r \geqq 0, 1, \ldots\}$. For some of the algorithms, under certain conditions, it can be shown that the sequence converges to a KKT point for the original problem. Unfortunately, there is no guarantee that a KKT point will be a local minimum, and our results point out that in general, checking whether or not it is a local minimum may be a hard problem.

There are several algorithms in the nonlinear programming literature which are based purely on the first order necessary conditions for a local minimum. These algorithms never use the objective value to guide them towards more desirable points. Instead, they concentrate purely on finding a solution to the system of first order necessary conditions. The class of complementary pivot methods or simplicial methods for NLP [1, 5, 7, 10] are examples of algorithms in this class (these algorithms convert the system of first order necessary conditions into a Kakutani fixed point problem, which is then solved by complementary pivoting on a triangulation of the space). These algorithms may at best lead to a KKT point. However, since the objective value is never even computed at any point, we do not have any circumstantial or neighborhood information that the KKT point obtained may be a local minimum. Thus, these algorithms may not be desirable algorithms to use on nonconvex NLPs.

Descent algorithms for NLPs are iterative algorithms with the property that the sequence of points generated is a descent sequence: either the objective function, or a measure of the infeasibility of the current solution, or some merit function which is a combination of both, strictly decreases along the sequence. Given the current point x^r, these algorithms generate $y^r \neq 0$ such that $x^r + \lambda y^r$, $\lambda \geqslant 0$, is a descent direction for the functions discussed above. The next point in the sequence, x^{r+1}, is usually taken to be the one that approximately minimizes the objective (or merit) function on the half-line $\{x^r + \lambda y^r: \lambda \geqslant 0\}$, and is obtained by using a line minimization algorithm. For general nonconvex problems these methods suffer from the same difficulties: they cannot theoretically guarantee that the point obtained at termination is a local minimum. However, it seems reasonable to expect that a solution obtained through a descent process is more likely to be a local minimum, than a solution based purely on necessary optimality conditions. Thus, a suitable goal for algorithms for solving nonconvex NLPs seems to be a descent sequence converging to a KKT point. Several descent algorithms in the nonlinear programming literature do reach this goal, which suggests that descent algorithms may be the most desirable practical algorithms for tackling nonconvex NLPs.

One final note. Nowadays the probabilistic analysis of various aspects of optimization algorithms is a popular area of study. Consider the case where $D=(d_{ij})$ is a random square matrix of order n with unit diagonal elements, and with a probability distribution for off-diagonal entries which is symmetric around 0, and so the marginal expectation of each d_{ij} is zero $(i \neq j)$. Simple instances of this occur if each off-diagonal d_{ij} is a random variable independent and identically and uniformly distributed on the interval $[-1, +1]$; or when the vector of off-diagonal entries in each column of D is generated by the uniform distribution on the boundary of the unit sphere in $\mathbb{R}^{n-1}$ with its center at the origin. Such probabilistic models have been used extensively in the study of the average computational complexity of complementary and simplex-type pivot methods for linear programming. Here is a research problem. Under this probabilistic model, calculate the probability, q, that 0 is not a local minimum for the function $h(u)$ defined in (15). This q is the probability that the answer to Problem 11 in Section 4 is "yes".

The probability q is a measure on the possibility that existing NLP algorithms reach a wrong conclusion for problems generated by the above probabilistic mechanism. Our suspicion is that $1-q$ is small.

Acknowledgement

We thank the referee, associate editor, Editor, S.M. Pollock, and R. Saigal for help in improving the presentation of this paper.

References

[1] E. Allgower and K. Georg, "Simplicial and continuation methods for approximating fixed points and solutions to systems of equations," *SIAM Review* 22 (1980) 28–85.
[2] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques* (Wiley, New York, 1968).
[3] M.R. Garey and D.S. Johnson, *Computers and Intractability, A Guide to the Theory of NP-completeness* (Freeman, New York, 1980).
[4] O.L. Mangasarian, *Nonlinear Programming* (McGraw-Hill, New York, 1969).
[5] O.H. Merrill, "Application and extensions of an algorithm that computes fixed points of upper semicontinuous point to set mappings," Ph.D. Dissertation, University of Michigan, Ann Arbor, MI (1972).

[6] K.G. Murty, *Linear Programming* (Wiley, New York, 1983).
[7] K.G. Murty, *Linear Complementarity, Linear and Nonlinear Programming* (Heldermann-Verlag, West Berlin, 1987), to appear.
[8] K.G. Murty, *Linear and Combinatorial Programming* (Krieger Publishing Company, Malabar, FL, 1976).
[9] A. Schrijver, *Theory of Linear and Integer Programming* (Wiley-Interscience, New York, 1986).
[10] M.J. Todd, *The Computation of Fixed Points and Applications* (Springer-Verlag, Berlin, Heidelberg, New York, 1976).