ALGORITHMS FOR COMPUTER AIDED DESIGN OF CONTROL SYSTEMS BY THE METHOD OF INEQUALITIES

by

E. Polak and D. Q. Mayne

Memorandum No. UCB/ERL M79/48

26 July 1979

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

ALGORITHMS FOR COMPUTER AIDED DESIGN OF CONTROL SYSTEMS BY THE METHOD OF INEQUALITIES

E. Polak

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

D. Q. Mayne

Department of Computing and Control
Imperial College of Science and Technology
London SW7 2BZ, England

ABSTRACT

This paper collects a number of scattered results on the solution of both finite and infinite systems of inequalities into a unified whole. In particular, it presents two methods which solve finite systems of inequalities in a finite number of iterations and which use outer approximations techniques for infinite systems of inequalities.

Research sponsored by the National Science Foundation Grant ENG73-08214-A01, the Joint Services Electronics Program Contract F44620-76-C-0100 and the National Science Foundation (RANN) Grant ENV76-04274.

I. Introduction

It has been shown that many design problems, including that of designing controllers for multivariable linear and nonlinear systems, can be transcribed into the problem of satisfying nonlinear inequality constraints [1,2,3]. However, design specifications frequently result in surprisingly difficult inequalities (functional inequalities) for which standard algorithms are inadequate. Examples of functional inequalities in the time domain include (hard) constraints on outputs, states, control magnitudes, control rates etc. in response to standard inputs (e.g. step inputs); examples in the frequency domain include stability constraints on Nyquist locus and constraints on the closed loop frequency response; examples in parameter space include integrity (e.g. eigenvalues of closed-loop system must lie in open left half plane for all values of a parameter vector p in a set P). All these constraints can be expressed in the canonical form $g(x) \triangleq \max\{\phi(x,\alpha) \mid \alpha \in \mathcal{A}\} \leq 0$ where x is the design vector and α represents time t or frequency ω or parameter p etc. These constraints are very complex as they are essentially infinite dimensional. An algorithm for satisfying such inequalities is presented and it is shown that the efficient operation of the algorithm requires highly efficient sub-algorithms for finding a point x which satisfies a *finite* number of inequality constraints. Two newly developed algorithms for solving this latter problem are described and their performance evaluated by means of examples.

2. Outer Approximation Methods for Infinite Systems of Inequalities

To illustrate how cumbersome inequalities arise, let us consider the design of a simple feedforward controller, with transfer function $G(x,s)$, where x are the design parameters. First we are bound to have simple inequalities of the form $x \geq 0$. Second, the requirements of stability, as expressed by the Lienard-Chipart test, say, result in a system of algebraic inequalities $f^j(x) \leq 0$, $j = 1,2,\ldots,\ell$. Third, frequency response requirements may lead to inequalities of the form $g^k(x,\omega) \leq 0$ for all $\omega \in [\omega_0,\omega_f]$, $k = 1,2,\ldots,m_\omega$, and step response requirements (e.g. on peak overshoot) may lead to inequalities of the form $h^\ell(x,t) \leq 0$ for all $t \in [t_0,t_f]$, $\ell = 1,2,\ldots,m_t$, and finally it may be necessary to ensure that the system performance remains within specifications not only for the normal value $\hat{x}$ of the design but also for all realizations within tolerances i.e., for all $x = \hat{x} + \tau$, $\tau \in T$. Eventually, when one collects all of these inequalities one obtains a system of inequalities of the form

$$g^j(x) \leq 0 \quad j = 1,2,\ldots,\ell \tag{1}$$

$$\phi^k(x,\alpha) \leq 0 \quad \forall \alpha \in \mathcal{A}^k, \quad k = 1,2,\ldots,m \tag{2}$$

where the $g^j : \mathbb{R}^n \to \mathbb{R}^1$ and $\phi^k : \mathbb{R}^n \times \mathbb{R}^{n_k} \to \mathbb{R}^1$ are all continuously differentiable. Obviously, the major source of difficulty is caused by the inequalities of the form (2). Hence, without great loss of generality, but with a substantial gain in notational simplicity, the general method for solving (1)-(2) can be explained by considering the simplest case of (1), (2), viz, the system

$$\phi(x,\alpha) \leq 0, \ \alpha \in \mathcal{A} \tag{3}$$

where $\phi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^1$ is continuously differentiable and $\mathcal{A} \subset \mathbb{R}^m$ is a compact set.

In [9] we find algorithms for solving minimization problems with constraints of the form (3). We can extract from these algorithms a method for solving infinite systems of inequalities, such as (3). The basic idea is to decompose the solution of (3) into the solution of a possibly infinite sequence of <u>finite</u> systems of inequalities of the form

$$\phi(x,\alpha) \leq 0, \ \alpha \in \mathcal{A}_k, \ k = 0,1,2,\ldots \tag{4}$$

with $\mathcal{A}_k$ a finite (discrete) subset of $\mathcal{A}$. Obviously, for such a scheme to be efficient, we must have very rapid and finite subprocedures for solving (4) (with k fixed). Such subalgorithms will be described in the next section. First we must address ourselves to the construction of the $\mathcal{A}_n$ and examine the consequences of the scheme we are about to propose. Let $\psi : \mathbb{R}^n \to \mathbb{R}^1$ be defined by

$$\psi(x) \triangleq \max_{\alpha \in \mathcal{A}} \phi(x,\alpha) \tag{5}$$

then we see that x satisfies (3) if and only if $\psi(x) \leq 0$. Quite obviously, $\psi(x)$ is very expensive to compute, even approximately. We shall need to compute it approximately with progressively greater precision, as the computation progresses. Let $\ell : \mathbb{N} \to \mathbb{N}$ ($\mathbb{N} = \{1,2,3,\ldots\}$) be strictly monotonically increasing. If we apply $\ell(k)$ iterations of an optimization algorithm to the evaluation of (5), we get an α_k (more precisely $\alpha_{\ell(k)}$) and an approximate value of $\psi(x)$, which we denote by $\psi_k(x) = \phi(x,\alpha_k)$. (We assume initialization in a compact set for α).

We shall assume that our algorithm has the property that $|\psi_k(x) - \psi(x)| \to 0$ as $k \to \infty$, uniformly with x in any large bounded set. Furthermore, we shall need a double subscripted sequence, $\{\varepsilon_{jk}\}_{k \geq j}$ which satisfies the following hypotheses:

<u>H1</u>: $\varepsilon_{kk} = 0$ for all k and $\varepsilon_{jk} > 0$ for all $k > j$;

<u>H2</u>: $\varepsilon_{jk} \nearrow \hat{\varepsilon}_j > 0$ as $k \to \infty$, uniformly in j;

<u>H3</u>: $\hat{\varepsilon}_j \searrow 0$ as $j \to \infty$.

As we shall later see, it is advantageous to keep $\hat{\varepsilon}_j$ as large as possible for as long as possible. A typical such sequence is defined by $\varepsilon_{jk} = 100[(\frac{1}{j+1})^{1/10} - (\frac{1}{k+1})^{1/10}]$, $k \geq j$.

<u>Master Algorithm</u>

<u>Data</u>: $\mathcal{A}_0 \subset \mathcal{A}$ a discrete set. A double subscripted sequence $\{\varepsilon_{jk}\}_{k \geq j}$ satisfying H1 - H3.

<u>Step 0</u>: Set $k = 0$.

<u>Step 1</u>: Compute an x_k such that

$$\phi(x_k,\alpha) \leq 0 \text{ for all } \alpha \in \mathcal{A}_k \tag{6}$$

<u>Step 2</u>: Compute $\psi_k(x_k)$ and $\alpha_k \in \mathcal{A}$ such that $\psi_k(x_k) = \phi(x_k,\alpha_k)$.

<u>Step 3</u>: If $\psi_k(x_k) \leq 0$, set $x_{k+1} = x_k$, set $\mathcal{A}_{k+1} = \mathcal{A}_k$, set $k = k + 1$ and go to Step 2. Else proceed to Step 4.

<u>Step 4</u>: Construct

$$\mathcal{A}_{k+1} = \{\alpha_k\} \cup \{\alpha_j \in \mathcal{A}_k | \phi(x_j,\alpha_j) \geq \varepsilon_{jk}\} \tag{6a}$$

Set $k = k + 1$ and go to Step 1. ¤

Before we proceed to analyze this algorithm we note the effect of the behavior of the sequence ε_{jk}: for any given j, $\phi(x_j,\alpha_j) \geq \varepsilon_{jk}$ may hold for $k = j$ and a few more iterations, but as k increases, eventually $\phi(x_j,\alpha_j) < \varepsilon_{jk}$ is quite likely to occur and j is dropped from $\mathcal{A}_{k+1}$ and all subsequent $\mathcal{A}_\ell$, $\ell = k+2,\ldots$. This device tends to keep the cardinality of $\mathcal{A}_k$ small, enhancing the rapidity with which the points x_k are computed.

<u>Theorem 1</u>: Let $\{x_k\}_{k=0}^{\infty}$ by any sequence constructed by the Master Algorithm. Then any accumulation point $\hat{x}$ of $\{x_k\}_{k=0}^{\infty}$ satisfies $\psi(\hat{x}) \leq 0$.

<u>Proof</u>: Let $\{x_k\}_{k=0}^{\infty}$ be a sequence constructed by the Master Algorithm. We distinguish two cases. First suppose that for all $k \geq k_0$ the algorithm cycles in the loop defined by Step 2 - Step 3. Then we have $x_k = x_{k_0}$ for all $k \geq k_0$ and $\psi_k(x_{k_0}) \leq 0$ for all $k \geq k_0$. But by assumption $\psi(x_{k_0}) = \lim_{k\to\infty} \psi_k(x_{k_0})$ and hence $\psi(x_{k_0}) \leq 0$.

Next, we suppose that the algorithm does not cycle indefinitely between Steps 2 and 3 and that $x_k \to \hat{x}$ as $k \to \infty$, $k \in K \subset \mathbb{N}$. To obtain a contradiction, we assume that $\psi(\hat{x}) > 0$. Then, since $\psi(\cdot)$ is continuous (see B.3.20 in [10]) $\psi(x_k) \to \psi(\hat{x})$ as $k \to \infty$, $k \in K$, and therefore, since $\psi_k(x_k) \to \psi(x_k)$ as $k \to \infty$, $k \in K$, there exists a k_1 such that $\psi_j(x_j) \geq \psi(x)/2 \geq \hat{\varepsilon}_j \geq \varepsilon_{jk}$ for all $j \in K$, $j \geq k_1$ and $k \geq j$. But this implies that $\alpha_j \in \mathcal{A}_k$ for all $j \in K$, $j \geq k_1$, and all $k \geq j + 1$. Now, by construction of x_k, $\phi(x_k,\alpha) \leq 0$ for all $\alpha \in \mathcal{A}_k$ and hence

$$\phi(x_k,\alpha_j) \leq 0 \quad \text{for all } j \in K,\ j \geq k, \quad \text{and } k \geq j + 1 \tag{7}$$

Now, $\mathcal{A}$ is compact and hence $\phi(\cdot,\alpha)$ is continuous, uniformly in $\alpha \in \mathcal{A}$. Therefore

$$|\phi(x_k,\alpha_j) - \phi(x_j,\alpha_j)| \to 0 \quad \text{as } j,k \to \infty \tag{8}$$

with $j,k \in K$ and $k \geq j + 1$. Combining (7) and (8), we obtain that

$$\overline{\lim_{j\in K}}\, \psi_j(x_j) \leq 0 \tag{9}$$

and, since $|\psi_j(x_j) - \psi(x_j)| \to 0$ as $j \to \infty$, and $\psi(x_j) \to \psi(\hat{x})$, as $j \to \infty$, we must have $\psi(\hat{x}) \leq 0$. But this contradicts our hypothesis and hence we are done. ⌑

Although Theorem 1 does not rule out the possibility that the cardinality of the sets $\mathcal{A}_k$ increases without bound, in practice it has been found that the cardinality of the sets $\mathcal{A}_k$ remains quite small, with $\mathcal{A}_k$ containing only a few points. Also, x_k is an excellent initial point for computing x_{k+1} by means of one of the algorithms to be described in the next section and hence the whole computation progresses quite efficiently.

3. Finite Algorithms for Finite Systems of Inequalities

Consider the system of inequalities

$$g^i(x) \leq 0 \quad j \in \underline{m} \tag{10}$$

where $\underline{m} = \{1,2,\ldots,m\}$ and the $g^j : \mathbb{R}^n \to \mathbb{R}^1$ are Lipschitz continuously differentiable. By analogy with (5), we adopt the notation

$$\psi(x) \triangleq \max_{j\in\underline{m}} g^j(x) \tag{11}$$

No confusion with (5) will arise since we need no more refer to the case of infinite inequalities. We shall need the following additional assumptions.

<u>H4</u>: There exists an $x^* \in \mathbb{R}^n$ such that $\psi(x^*) < 0$.

<u>H5</u>: $0 \notin \mathrm{co}\{\nabla g^j(x) \mid j \in I(x)\}$ for all x such that $\psi(x) \geq 0$, where

$$I(x) \triangleq \{j \in \underline{m} \mid g^j(x) = \psi(x)\} \tag{12}$$

and co denotes the convex hull of the set in question. ¤

The algorithms appear to work also under the weaker assumption

<u>H6</u>: $\frac{\partial g(x)}{\partial x}^T g(x)_+ = 0$ if and only if $g(x)_+ = 0$ where $g(x)_+ \in \mathbb{R}^m$ is a vector with components

$$[g(x)_+]^j \triangleq \max\{0, g^j(x)\} \quad j \in \underline{m} \tag{13}$$

¤

It is well known (see e.g. Ch. 4 in [10]) that when H4 and H5 are satisfied, it is possible to find a point $\bar{x}$ solving (10) in a finite number of iterations of any of a number of methods of feasible directions. A particularly simple version of such an algorithm is developed in [11][†] on the basis of the phase 1 Topkis-Veinott method [10]:

<u>Algorithm 1</u>:

<u>Data</u>: $x_0 \in \mathbb{R}^n$

<u>Parameters</u>: $\alpha \in (0,1)$, $\beta \in (0,1)$, $\gamma > 0$.

<u>Step 0</u>: Set $i = 0$.

<u>Step 1</u>: If $\psi(x_i) \leq 0$, stop. Else, compute a direction vector h_i as a solution of the linear program (Topkis-Veinott [10])

$$\theta_{TV}(x_i) \triangleq \min_{\|h\|_\infty \leq \gamma} \max_{j \in \underline{m}} g^j(x_i) + \langle \nabla g^j(x_i), h \rangle - \psi(x_i) \tag{14}$$

[†]Set the cost $f(x) \equiv -\infty$ and $\nabla f(x) \equiv 0$ in the algorithms in [11].

<u>Step 2</u>: Compute the step length $\lambda_i = \beta^{k_i}$, by the Armijo rule [10]:

$$k_i = \arg\max_{k\in\mathbb{N}} \{\beta^k | \psi(x_i+\beta^k h_i) - \psi(x_i) \le \beta^k\alpha\, \theta_{TV}(x_i)\} \tag{15}$$

<u>Step 3</u>: Set $x_{i+1} = x_i + \beta^{k_i} h_i$, set $i = i + 1$ and go to Step 1. ¤

The hypothesis H5 ensures that $\theta_{TV}(x_i) < 0$ and $h_i \neq 0$ for all $x_i \notin F \triangleq \{x|g^j(x) \le 0,\ j \in \underline{m}\}$. The hypothesis H4 ensures that the minimizer of $\psi(\cdot)$ lies in the interior of F and hence that the boundary of F is crossed after a finite number of iterations. To be quite precise, the relevant properties of Algorithm 1 are as follows:

<u>Theorem 1 [11]</u>: Suppose that H4 and H5 are satisfied. Then, either Algorithm 1 constructs an $x_i \in F \triangleq \{x|g^j(x) \le 0,\ j \in \underline{m}\}$ in a finite number of iterations, or it constructs an infinite sequence $\{x_i\}$ which has no accumulation points (and hence is unbounded). The following result is obvious. ¤

<u>Corollary 1</u>: If the set $\{x|\psi(x) \le \psi(x_0)\}$ is compact, Algorithm 1 finds a solution x_s to (10) finitely. ¤

An algorithm such as Algorithm 1 (in [11] we also find more complex, but also more efficient algorithms in this class), finds a solution $x_s \in F$ quite rapidly when the set F is "fat." It can become quite slow when the set F is "narrow." On the other hand, appropriate versions of Newton's method (see [4,5]) always converge very rapidly, with only one hitch: they do not converge finitely. In the rest of this section we shall show how Newton's method can be combined either with Algorithm 1 or with a boundary shifting technique to obtain a very rapid method for solving (10) finitely. The simplest version of Newton's method for solving (10) was proposed by Robinson [4]. Given x_i, it solves the program

$$\min\{\|v\|_k \,|\, g(x_i) + \frac{\partial g(x_i)}{\partial x} v \leqq 0\}, \quad k = 2,\infty \cdot \tag{16}$$

for a v_i and sets

$$x_{i+1} = x_i + v_i, \quad i = 0,1,2,\ldots \tag{17}$$

where $g(x) \triangleq (g^1(x),\ldots,g^m(x))^T$. When $\|v\|_\infty \triangleq \max_j |v^j|$ is used, (16) is a linear program, when $\|v\|_2^2 \triangleq \sum_{j=1}^m (v^j)^2$, is used, (16) is a quadratic program. Either can be used, the trade off being that while the L_2 norm gives better computational behavior, it makes (16) harder to solve. It was shown by Robinson that under H5, if $\psi(x_0)$ is sufficiently small, then $\{x_i\}$ constructed according to (16) and (17) converges with root rate 2 to an $\hat{x}$ satisfying $\psi(\hat{x}) \leq 0$, i.e. for some $M > 0$, $\delta \in (0,1)$, $\|x_i - \hat{x}\| \leq M\delta^{2^i}$, $i = 0,1,2,\ldots$.

A stabilized version of Robinson's algorithm was proposed by Mukai and Polak in [5].. Their version converges globally and defaults to an Armijo type gradient method [10] for minimizing $\frac{1}{2}\|g(x)_+\|^2$ when Newton's method fails. The vector $g(x)_+ \in \mathbb{R}^m$ is defined for all $x \in \mathbb{R}^n$ by

$$[g(x)_+]^j = \max\{0, g^j(x)\}, \; j \in \underline{m}. \tag{18}$$

<u>Stabilized Newton Method [5]</u>

<u>Parameters</u>: $\alpha \in (0,1/2)$, $\beta \in (0,1)$, $L >> 1$.

<u>Data</u>: $x_0 \in \mathbb{R}^n$

<u>Step 0</u>: Set $i = 0$.

<u>Step 1</u>: Stop if $\psi(x_i) \leq 0$. Else, solve the program

$$\min\{\|v\|_k^2 \,|\, g(x_i) + \frac{\partial g(x_i)}{\partial x} v \leqq 0\} \ , \tag{19}$$

for v_i (where $\|\cdot\|$ is either ℓ_2 or ℓ_∞ norm).

Step 2: If v_i exists and $\|v_i\| \leq L$, set $p_i = v_i$, else set $p_i = -\frac{\partial g(x_i)^T}{\partial x} g(x_i)_+$.

Step 3: Compute the smallest integer $k_i \geq 0$ such that

$$\|g(x_i+\beta^{k_i} p_i)_+\|^2 \leq (1-2\alpha\beta^{k_i})\|g(x_i)_+\|^2 \tag{20}$$

Step 4: Set $x_{i+1} = x_i + \beta^{k_i} p_i$, set $i = i + 1$ and go to step 1. ⌑

Theorem 2 [5]: Suppose that H5 holds. If the stabilized Newton method constructs an infinite sequence $\{x_i\}$, then, either this sequence has no accumulation points or it converges with root rate 2 to an $\hat{x}$ such that $g(\hat{x}) \leq 0$. ⌑

First we show how Robinson's Newton method can be combined with Algorithm 1 to produce a method which is both rapid and which also terminates finitely. The idea is to take one iteration of Newton's method followed by one iteration of Algorithm 1, with certain precautions added to ensure overall convergence.

Algorithm 2 [7]:

Parameters: $\alpha \in (0,1/2)$, $\beta \in (0,1)$, $\gamma > 0$, $L >> 1$.

Data: $x_0 \in \mathbb{R}^n$.

Step 0: Set $i = 0$.

Step 1: If $\psi(x_i) \leq 0$, stop.

Step 2: Solve for $p(x_i)$ the Newton program

$$\min\{\|p\| \,|\, g(x_i) + \frac{\partial g(x_i)}{\partial x} p \leqq 0\} \tag{21}$$

where $\|\cdot\|$ is either the ℓ_∞ or ℓ_2 norm. If $p(x_i)$ exists and $\|p(x_i)\| \leq L$, solve for $p'(x_i)$ the modified Topkis-Veinott descent direction program

$$\min_{\|p'\|\leq\gamma}\ \max_{j\in\underline{m}} \{g^j(x_i) + \langle\nabla g^j(x_i), p(x) + p'\rangle\} \tag{22}$$

and set $p_i = p(x_i) + p'(x_i)$. If $p(x_i)$ does not exist or $\|p(x_i)\| > L$, solve for $p''(x_i)$ the Topkis-Veinott descent direction program

$$\min_{\|p''\|\leq 1}\ \max_{j\in\underline{m}}\{g^j(x_i) + \langle\nabla g^j(x_i), p''\rangle\} \tag{23}$$

and set $p_i = p''(x_i)$.

Step 3: Determine the smallest integer $k_i \geq 0$ such that (c.f. (15))

$$\psi(x_i+\beta^{k_i}p_i) - \psi(x_i) \leq \alpha\beta^{k_i}[\max_{j\in\underline{m}}\{g^j(x_i) + \langle\nabla g^j(x_i), p_i\rangle\} - \psi(x_i)] \tag{24}$$

Step 4: Set $x_{i+1} = x_i + \beta^{k_i}p_i$, set $i = i + 1$ and go to Step 1. ¤

The qualitative convergence properties of Algorithm 2 are identical to those of Algorithm 1, as is shown in the theorem below, reproduced from [7].

Theorem 3: Suppose that H4 and H5 are astisfied. Then, either Algorithm 2 constructs an $x_s \in F \triangleq \{x | g^j(x) \leq 0,\ j \in \underline{m}\}$ in a finite number of iterations, or it constructs an infinite sequence $\{x_i\}$ which has no accumulation points. ¤

Corollary 1 holds true also for Algorithm 2.

An alternative approach to pushing Newton's method "over the brink" in a finite number of iterations was proposed in [8]. The idea is quite simple: substitute for the system (10) the system

$$g^j(x) + \varepsilon \leq 0 \quad j \in \underline{m}, \tag{25}$$

with $\varepsilon > 0$ such that (25) has a solution and apply Newton's method to that system. In the process, Newton's method finds a solution $\hat{x}$ to the original system (10) in a finite number of iterations. The only difficulty with this approach is that one does not know in advance for what values of $\varepsilon > 0$ the system (25) has a solution. The idea proposed in [8] for finding a satisfactory $\varepsilon > 0$ is as follows. If $\varepsilon > 0$ is small enough and x_0 is such that $\psi(x_0)$ is small enough, then, the sequence constructed by Newton's method [8] satisfies

$$\psi(x_i) \leq -\varepsilon + M\delta^{2^i} \quad i = 0,1,2,\ldots \tag{26}$$

for some $M > 0$ and $\delta \in (0,1)$, both independent of ε. Hence, if for $k = 0,1,2,\ldots$, we let $\varepsilon = \gamma_1^k\varepsilon_0$, with $\gamma_1 \in (0,1)$, and $i = \ell_k$, where ℓ_k is an integer such that $\ell_k \nearrow \infty$ as $k \to \infty$, we get for $z_k = x_{\ell_k}$ in (26):

$$\psi(z_k) \leq -\gamma_1^k\varepsilon_0 + M\delta^{2^{\ell_k}} \leq 0 \tag{27}$$

for k finite, sufficiently large. With a little more work, it can be shown that if an additional safeguard (see (30), below) is added, and, for each value of $\varepsilon = \varepsilon_k \overset{\Delta}{=} \gamma_1^k\varepsilon_0$, Newton's method is applied to (25) for at least ℓ_k iterations, then the inequality (26) eventually becomes valid and by (27), the process must be finite. The exact structure of the algorithm in [8] is as follows.

Algorithm 3.

Parameters: $\alpha \in (0,1/2)$, $\beta \in (0,1)$, $L >> 1$, γ_1, $\gamma_2 \in (0,1)$, $\delta \in (0,1)$, integers $\{\ell_k\}_{k=0}^{\infty}$ such that $\ell_{k+1} > \ell_k$ for all k.

Data: $z_0 \in \mathbb{R}^n$, $\varepsilon_0 > 0$.

Step 0: Set $k = 0$.

Step 1: (Initialization of Newton's method [5] for (25).) If $\psi(z_k) \leq 0$, stop. Else, set $i = 0$, $x_0 = z_k$, $\varepsilon = \varepsilon_k$.

Step 2: Solve the linear or quadratic program

$$\min\{\|v\| \mid g^j(x_i) + \langle \nabla g^j(x_i), v\rangle + \varepsilon \leq 0, \quad j \in \underline{m}\} \tag{28}$$

for v_i.

Step 3: If v_i exists and $\|v_i\| \leq L$, set $p_i = v_i$. Else set $p_i = -\frac{\partial g(x_i)^T}{\partial x} g_\varepsilon(x_i)_+$ where $g_\varepsilon(x)_+ \in \mathbb{R}^m$ is defined by $g_\varepsilon^\ell(x)_+ = \max\{0, g^\ell(x) + \varepsilon\}$, $\ell \in \underline{m}$ (negative gradient direction of $\frac{1}{2}\|g_\varepsilon(x_i)_+\|^2$).

Step 4: (Armijo step length rule) Compute the smallest integer $j \geq 0$ such that

$$\|g_\varepsilon(x_i + \beta^j p_i)_+\|^2 \leq (1 - 2\alpha\beta^j)\|g_\varepsilon(x_i)_+\|^2 \tag{29}$$

Step 5: Set $x_{i+1} = x_i + \beta^j p_i$.

Step 6: (Test if it is time to reduce ε.) If $i \geq \ell_k$ and

$$\psi(x_{i+1}) \leq \gamma_1 \psi(x_i) + (1 - \gamma_1)(\sqrt{m} - 1)\varepsilon_k \tag{30}$$

set $z_{k+1} = x_{i+1}$, $\varepsilon_{k+1} = \gamma_2 \varepsilon_k$, set $k = k + 1$ and go to Step 1. Else set $i = i + 1$ and go to Step 2. ⌑

Theorem 4 [8]: Either Algorithm 3 constructs a $z_k \in \mathbb{R}^n$ satisfying (10) in a finite number of iterations, or the sequence $\{z_k\}$ is infinite and has no accumulation points. ⌑

4. Computational Aspects

Algorithm 1 will work even when the assumptions for Newton's method are not satisfied and hence is the most robust of the three. Also, when the set $\{x | \psi(x) \leq 0\}$ is "fat," Algorithm 1 makes rapid progress, but it can become quite slow when that set is "thin." Algorithm 2 combines these robustness aspects of Algorithm 1 with the added speed of convergence inherent in Newton's method. However, this is done at the price of having to solve an additional linear program at each iteration. Algorithm 3 totally depends on the properties of Newton's method. Thus, when the required assumptions are satisfied, it is as fast as Algorithm 2 and somewhat more efficient, because only one linear (or quadratic) program needs to be solved at each iteration. However, it requires somewhat more skill to use, because of the need to select ε_0 and γ_1, γ_2.

To illustrate the ease with which systems of inequalities can be solved by the algorithms in this paper, we present the examples, below, which were solved using Algorithm 2. Algorithm 3 gave basically identical results except for example 3 on which it failed because the assumptions for Newton's method are grossly violated.

Example 1. The feasible set consists of infinitely many squares centered at $((2m-1/2)\pi, 2n\pi)$ for all integer values of m,n. The set is defined by:

$$\sin x^1 \leq 0$$

$$-\cos x^2 \leq 0$$

with $x_0 = (1,2)$, a feasible point (-3,0416,1.4708) was located in one step (a (modified) Newton step).

<u>Example 2</u>. The feasible set consists of a pair of squares centered at $(-\pi/2,0)$ and $(3\pi/2,0)$, and is defined by:

$$\sin x^1 \leq 0$$

$$-\cos x^2 \leq 0$$

$$x^1 - 3\pi \leq 0$$

$$x^2 - \pi/2 \leq 2$$

$$-x^1 - \pi \leq 0$$

$$-x^2 - \pi/2 \leq 0$$

With $x_0 = (0,75)$, a feasible point $(-3.0416,1.4708)$ was located in four iterations. The sequence of points generated were $(1,74)$, $(1,75)$, $(1,72)$, $(-3.0416,1.4708)$, the first three steps being first order, and the final one a (modified) Newton step.

<u>Example 3</u>. The feasible set is defined by:

$$(0.999)^2 \leq (x^1)^2 + (x^2)^2$$

With $x_0 = (0,15)$ a feasible step was located in four iterations, all steps being of the (modified) Newton type.

<u>Example 4</u>. The feasible set is narrow crescent defined by:

$$(x^1-1/2)^2 + (x^2-1)^2 \leq 0.25$$

$$-(x^1-1/2)^2 - (x^2-1.1)^2 \leq 0.26$$

$$x^2 - 1 \leq 0$$

Starting from an initial point $(0.5,-6.0)$ a feasible point is located a five iterations, all of the (modified) Newton type.

Example 5. This problem is based on one suggested by Powell. The vector coordinates $(x^1,0)$, (x^2,x^3) of two vertices of a triangle (the third being $(0,0)$) and the centers (x^4,x^5), (x^6,x^7) of two discs of unit radius. A point x is feasible if the triangle has a specified area a and the two discs lie inside the triangle and do not overlap each other. These constraints can be specified as:

$$x^1 \geq 0$$

$$x^3 \geq 0$$

$$(x^4-x^6)^2 + (x^5-x^7)^2 - 4 \geq 0$$

$$x^i - 1 \geq 0, \quad i = 5, 7$$

$$\frac{x^3x^6-x^2x^{i+1}}{[(x^2)^2+(x^3)^2]^{1/2}} - 1 \geq 0, \quad i = 4,6$$

$$\frac{(x^2-x^1)x^{i+1}+(x^1-x^i)x^3}{[(x^3)^2+(x^2-(x^1)^2)^2]^{1/2}} - 1 \geq 0, \quad i = 4, 6$$

$$a - 1/2\, x^1x^3 \geq 0$$

With x_0 = ((3), (0,2),(-1.5,1-5),(5,0)) and a = 12, a feasible point ((5.8381),(0.4448,4.1062),(1.250,2.3237),(2.8434,1.0505)) was achieved in six iterations. The minimum value for a has been obtained by Powell as 11.6569.

5. Conclusion

We have presented three efficient method for solving both finite and infinite systems of inequalities. We hope that these methods will be of significant use to those who design systems by the method of inequalities.

References

[1] N. Zakian and U. Al-Naibk "Design of Dynamical and Control Systems by the Method of Inequalities," Proc. IEE, Vol. 120, No. 11, 1973, pp. 1421-1427.

[2] J. W. Bandler and H. L. Abdul-Malek, "Advances in the Mathematical Programming Approach to Design Centering, Tolerancing and Tuning," Proc. 1978 JACC, Philadelphia, Oct. 14-21, 1978, pp. 329-344.

[3] E. Polak and A. Sangiovanni-Vincentelli, "On Optimization Algorithms for Engineering Design Problems in the Distributed Constraints, Tolerances and Tuning," Proc. 1978 JACC, Philadelphia, Oct. 19-21, 1978, pp. 344-353.

[4] S. M. Robinson, "Extension of Newton's Method to Method to Mixed Systems of Nonlinear Equations and Inequalities," Tech. Summ. Rep. 1161, Math. Research Center, Univ. of Wisconsin, Madison, Wisconson, 1971.

[5] H. Mukai and E. Polak, "On the Use of Approximations in Algorithms for Optimization Problems with Equality and Inequality Constraints," SIAM. J. Numer. Anal., Vol. 15, No. 4, 1978, pp. 674-693.

[6] S. M. Robinson, "Bounds for Error in the Solution Set of a Perturbed Linear Program," Linear Algebra and Its Applications, Vol. 6, 1973, pp. 69-81.

[7] D. Q. Mayne, E. Polak and A. J. Heunis, "Solving Nonlinear Inequalities in a Finite Number of Iterations," Memorandum No. UCB/ERL M79/12, University of California, Berkeley, California.

[8] E. Polak and D. Q. Mayne, "On the Finite Solution of Nonlinear Inequalities," Memorandum No. UCB/ERL M78/30, University of California, Berkeley, California.

[9] C. Gonzaga and E. Polak, "On Constraint Dropping Schemes and Optimality Functions for a Class of Outer Approximations Algorithms," SIAM J. Control & Opt., Vol. 17, No. 4, 1979.

[10] E. Polak, Computational Methods in Optimization, Academic Press, 1971.

[11] E. Polak, R. Trahan and D. Q. Mayne, "Combined Phase I-Phase II Methods of Feasible Directions," Math. Programming, Vol. 16, No. 4, 1979.