On-line Solution of Linear Programs Using Sublinear Functions

JOHN R. BIRGE
Industrial and Operations Engineering
The University of Michigan

ROGER J-B. WETS
Department of Mathematics
University of California at Davis

Technical Report #86-25

July 1986

On-line Solution of Linear Programs Using Sublinear Functions

John R. Birge*
Industrial and Operations Engineering
University of Michigan
Ann Arbor, MI 48109

Roger J-B. Wets*
Mathematics
University of California
Davis, CA 95616

Abstract:

When linear programs need to be solved many times with only changes in the right-hand side vector, approximation by sublinear functions can provide solutions very efficiently. This paper presents this sublinear approximation and its properties. The method may be especially useful if implemented on parallel processors.

Keywords: linear programming, sublinear functions, real-time optimization, parallel processing.

*This research has been partially supported by the National Science Foundation.

1. Introduction

Many uses of linear programming require the repeated solution of a single linear programming model with varying parameters. It is often important in these situations to solve the problems extremely quickly. The control of a ship or plane, for example, requires repeated optimization in real time.

A specific example of this need for precise, quick solutions is the model of ship control in restricted waters in Cuong [1980] and Al-Idrisi [1981]. In their model, several unknown parameters that depend on the waters must be identified to determine the ship's course. These parameters should be continuously updated as the ship moves. Rudder controls are also continuously determined to maintain the ship on a desired course. The parameters and controls can be found by solving a sequence of linear programs to minimize absolute deviation from observed data and the desired course. Only the right-hand side vector changes as observations are made on the ship's actual trajectory.

The goal of this paper is to present an efficient method for solving these sequences of linear programs with varying right-hand sides. We, therefore, wish to evaluate the function

$$\psi(t) := \inf_{x \in \mathbf{R}^n} \{cx \mid Ax = t,\ x \geq 0\}, \tag{1.1}$$

where $t \in \mathbf{R}^m$ represents the right-hand side parameters which may vary. We present a method for quickly evaluating or approximating $\psi(t)$ based on sublinear approximations. The method has especially strong potential for applications with parallel computation. We demonstrate this applicability by showing that our method's running time is lower even than average-care claims for general linear programming algorithms.

The method is based on an approximation scheme first developed for stochastic programs in Birge and Wets [1986a], and extended to a procedure for calculating upper bounds in Birge and Wets [1986b]. Its use with parallel processors was introduced in Wets [1985].

Section 2 presents useful properties of $\psi(t)$ and gives the basic sublinear function method. The accuracy and effort involved in the method are also discussed. Section 3 describes properties of the sublinear approximation and presents alternative strategies for its implementation. We conclude with a discussion of its running time on parallel processors in comparison with the simplex method and Karmarkar's projective method.

2. Sublinear Approximation Method

We assume that (1.1) is well-formulated so that the linear program is bounded below for all t. In this case, $\psi(t)$ is a <u>proper</u> (nowhere $-\infty$ and somewhere finite), <u>sublinear</u> (positively homogeneous and convex) function. Another useful property of ψ concerns its epigraph.

<u>PROPOSITION 2.1</u> The epigraph of ψ (epi ψ) is a convex polyhedral cone (i.e., ψ is a polyhedral convex function).

<u>Proof</u>: A point $(\alpha, t_1, \ldots, t_m) = (\alpha, t) \in \text{epi } \psi$ if and only if $\alpha \geq \psi(t)$. By duality, if $\psi(t) < +\infty$, then

$$\psi(t) = \max \{\pi t \mid \pi A \geq c\}, \tag{2.1}$$

where we need only consider the finite set of basic solutions, $\{\pi \text{ basic} \mid \pi A \geq c\} = \{\pi^1, \ldots, \pi^K\}$. We then have

$$\text{epi } \psi = \{(\alpha, t) \mid (\alpha, t)(-1, \pi^k) \leq 0, \ \ell = 1, \ldots, k\}. \tag{2.2}$$

This completes the proof. ∎

Since epi ψ is polyhedral, we can also write it as

$$\text{epi } \psi = \text{pos } \left[\binom{\alpha^\ell}{t^\ell},\ \ell = 1,\dots,L\right]$$

$$= \left\{ \binom{\alpha}{t} \in \mathbf{R}^{m+1} \;\middle|\; \binom{\alpha}{t} = \sum_{\ell=1}^{L} \mu^\ell \binom{\alpha^\ell}{t^\ell},\ \mu^\ell \geq 0,\ \ell = 1,\dots,L \right\}, \tag{2.3}$$

so that

$$\psi(t) = \inf\left\{\alpha \;\middle|\; \binom{\alpha}{t} = \sum_{\ell=1}^{L} \mu_\ell \binom{\alpha^\ell}{t^\ell},\ \mu_\ell \geq 0,\ \ell = 1,\dots,L\right\}. \tag{2.4}$$

This leads to our first fundamental approximation. Suppose

$$\left\{\binom{\alpha^s}{t^s} \in \text{epi } \psi,\ s = 1,\dots,S\right\}, \tag{2.5}$$

is a collection of vectors in $\mathbf{R}^{m+1}$. We then have that

$$\psi(t) \leq \inf\left\{\alpha \;\middle|\; \binom{\alpha}{t} = \sum_{s=1}^{S} \mu_s \binom{\alpha^s}{t^s},\ \mu_s \geq 0,\ s = 1,\dots,S\right\}. \tag{2.6}$$

The right-hand side of (2.6) is another sublinear function that approximates $\psi(t)$ from above. We show how to choose the vectors $\binom{\alpha^s}{t^s}$ so that the approximation is reasonable and the computation of the approximation (finding the infimum) is straightforward.

First note that a good approximation would be exact along the directions t^s, $s = 1,\dots,S$. Hence, we require that $\alpha^s = \psi(t^s)$, and assume that α^s and t^s are defined in this way in our discussion below. We assume that the effort to calculate these α^s values is small in comparison with the number of times $\psi(t)$ must be found. The solutions x^s such that $cx^s = \alpha^s$ are also stored, in general, so that a primal solution $x^* = \sum_{s=1}^{S} \mu^s x^s$ is obtained for each solution in (2.6). If the x^s solutions are not available, then the dual may be used to find x^* (see below).

The set of vectors $\{t^1,\dots,t^S\}$ must be sufficient for the approximation to be reasonable. For pos $A = \{y \mid Ax = y,\ x \geq 0\}$, we generally assume that

$$\text{pos }(t^1,\ldots,t^S) \supset \text{pos } A. \qquad (2.7)$$

To ensure (2.7), we often have

$$\text{pos }(t^1,\ldots,t^S) \supset \mathbf{R}^m. \qquad (2.8)$$

Conditions for the set to obtain equality in (2.6) are given in the following proposition.

PROPOSITION 2.2 If $\{t^1,\ldots,t^S\} \supset \{\lambda_1 A_{.1},\ldots,\lambda_n A_{.n}\}$ for $\lambda_i > 0$, $i=1,\ldots,n$, and $\alpha^s = \psi(t^s)$, $s=1,\ldots,S$, then inequality (2.6) is satisfied as an equality for all t.

Proof: Suppose $t^i = \lambda_i A_{.i}$, $i=1,\ldots,n$. Let $\psi(t) = \sum_{i=1}^n q_i x_i^*$. Note that, by the assumptions, $\alpha^i \leq \lambda_i q_i$. Since $\sum_{i=1}^n A_{.i} x_i^* = t$, we have $\sum_{i=1}^n t^i \left(\frac{x_i^*}{\lambda_i}\right) = t$.

Hence,

$$\inf \left\{\alpha \mid \binom{\alpha}{t} = \sum_{s=1}^{S} \mu_s (t^s),\ \mu_s \geq 0,\ s=1,\ldots,S\right\}$$

$$\leq \sum_{i=1}^n \left(\frac{x_i^*}{\lambda_i}\right) \alpha^i \leq \sum_{i=1}^n q_i x_i^* = \psi(t). \qquad (2.9)$$

Combining (2.9) and (2.6) yields the result. ∎

A difficulty in using the expression in (2.6) is that another optimization problem is required in computing the bound. Exact results could be obtained as in Proposition 2.2, but no clear time savings would be made. The situation is simplified if the expression in (2.6) has a unique solution for all $t \in \mathbf{R}^m$, i.e., if $t^1,\ldots,t^S$ form a positive linear basis for $\mathbf{R}^m$. Positive linear bases can be easily obtained from linear bases by including the negative vector of each vector in the linear basis. Let $D = [D^1,\ldots,D^m]$ form a linear basis for $\mathbf{R}^m$, then $[D,-D] = [D^1,\ldots,D^m, -D^1,\ldots,-D^m]$ form a positive linear basis for $\mathbf{R}^m$. This is the form we shall use for approximations of $\psi(t)$.

For each D^j used in the approximation, let $\delta_j^+ = \psi(D^j)$ and $\delta_j^- = \psi(-D^j)$. The sublinear approximation using basis D is defined as

$$\psi_D(t) = \inf \left\{\alpha \mid \binom{\alpha}{t} = \sum_{j=1}^{n} \mu_j^+\binom{\delta_j^+}{D^j} + \sum_{j=1}^{n} \mu_j^-\binom{\delta_j^-}{-D^j},\right.$$
$$\left. \mu_j^+, \mu_j^- \geq 0, \; j = 1, \ldots, n\right\}. \tag{2.10}$$

Note that $\psi(t) \leq \psi_D(t)$ from (2.6) and that

$$\psi(t) = \psi_D(t) \quad \text{if} \quad t = \pm D^j, \; j=1,\ldots,m.$$

The expression for $\psi_D(t)$ can also be simplified to

$$\psi_D(t) = \sum_{j=1}^{m} \psi_D^j(t), \tag{2.11}$$

where

$$\psi_D^j(t) = \begin{cases} \delta_j^+(D^{-1})_{j.}\, t & \text{if } (D^{-1})_{j.}\, t \geq 0, \\ -\delta_j^-(D^{-1})_{j.} t & \text{if } (D^{-1})_{j.}\, t \leq 0. \end{cases} \tag{2.12}$$

Computing $\psi_D(t)$, therefore, only requires multiplication of t by the rows of D^{-1} for each j, a sign check, multiplication by $\delta_j{}^{\pm}$ and addition of the ψ_D^j. This decomposition of computational effort leads to the parallel computation gains discussed in the next section. In general, $\psi_D(t)$ can be evaluated quickly given initial evaluation of $\delta_j{}^{\pm}$, j=1,...,m.

A single basis D does not in general provide good bounds on $\psi(t)$ for all t. A collection of bases, $\mathcal{D} = \{D(\nu), \nu = 1,\ldots,N\}$, can be used instead. The approximations derived from this collection are described in the next proposition.

PROPOSITION 2.3. Let $\{D(\nu), \nu=1,\ldots,N\}$ be a collection of linear bases of $\mathbf{R}^m$, then

$$\psi(t) \leq \text{co}\ \psi_{D(\nu)}\ (t) \leq \inf\ \psi_{D(\nu)}(t), \tag{2.13}$$

where

$$\text{co}\ \psi_{D(\nu)}\ (t) = \inf\left[\alpha \;\middle|\; \binom{\alpha}{(t)} \in \text{co(epi}\ \psi_{D(\nu)},\ \nu = 1,\ldots,N)\right] \tag{2.14}$$

and "co" denotes "convex hull."

<u>Proof</u>: Note that epi $\psi \supset$ epi $\psi_{D(\nu)}$ for $\nu=1,\ldots,N$, from (2.6). Hence, since epi ψ is convex, it contains the smallest convex set containing epi $\psi_{D(\nu)}$, $\nu=1,\ldots,N$, i.e.,

$$\text{epi}\ \psi \supset \text{co(epi}\ \psi_{D(\nu)},\ \nu=1,\ldots,N). \tag{2.15}$$

From (2.15), we obtain $\psi(t) \leq \text{co}\psi_{D(\nu)}(t)$. The other inequality, $\text{co}\Psi_{D(\nu)}(t) \leq \inf_{\nu} \psi_{D(\nu)}(t)$ follows from the definition in (2.14). ■

By taking the convex hull of the functions $\psi_{D(\nu)}$, the approximation can improve and approach the exact value of ψ. The inclusion of more bases in the collection, $\{D(\nu),\ \nu=1,\ldots,N\}$, also improves the solution. Conditions for a collection to obtain equality in (2.13) are given in the following proposition.

<u>PROPOSITION 2.4</u> Let the collection $\{D(\nu),\ \nu=1,\ldots,N\}$ contain all dual feasible bases in W(i.e., a basis B such that $\pi B=c_B$, $\pi A \leq c$), then

$$\psi(t) = \text{co}\ \psi_{D(\nu)}(t) = \inf_{\nu}\ \psi_{D(\nu)}\ (t). \tag{2.16}$$

<u>Proof</u>: Let $\psi(t) = cx^* = \pi^*t$ where $Bx^*_B = t$, $\pi^*B = c_B$, and $\pi^*A \leq c$. For $t^i = B^i$, $i = 1,\ldots,m$, we again have $\alpha^i \leq c_i$, so for $D(j) = B$, $\psi_{D(j)}(t) = \sum_{i=1}^{m} \alpha^i x^*_B(i) \leq cx^* = \psi(t)$. From (2.13), we obtain (2.16). ■

Using all dual feasible bases in $\mathcal{D}$ would generally not be efficient. The convex hull function can also be a computational burden on the sublinear approximation method. The next section discusses these implementation concerns and describes other properties of the sublinear approximation method.

3. Uses of the Approximation

To compute co $\psi_{D(\nu)}(t)$, the optimization problem in (2.14) must be solved. Since this problem may, in general, be difficult, the simpler bound of $\inf_\nu \psi_{D(\nu)}(t)$ is indicated. Another alternative that is especially useful when $\psi(t)$ is part of a higher level optimization problem is to use the conjugate function of co $\psi_{D(\nu)}$. The conjugate of ψ is by definition

$$\psi^*(a) = \sup_{t \in \mathbf{R}^m} [ut - \psi(t)]. \tag{3.1}$$

The following proposition provides the conjugate of $\psi_{D(\nu)}$.

PROPOSITION 3.1 The conjugate of $\psi_{D(\nu)}$ is given by

$$\psi_{D(\nu)}{}^*(u) = \begin{cases} 0 \text{ if } -\delta_j^- \le u(D(\nu))_{.j} \le \delta_j^+, \\ \qquad\qquad j = 1,\dots,m \\ +\infty \quad \text{otherwise.} \end{cases} \tag{3.2}$$

Proof: From the definition in (3.1) and for $D(\nu)$ a linear basis,

$$\psi_{D(\nu)}{}^*(u) = \sup_{\lambda \in \mathbf{R}^m} \left[uD(\nu)\lambda - \psi_{D(\nu)}\left(D(\nu)\lambda\right)\right]$$

$$= \sum_{j=1}^{M} \sup_{\lambda \in \mathbf{R}} \begin{matrix} \left(u(D(\nu))_{.j} - \delta_j^+\right)\lambda_j & \text{if } \lambda_j \ge 0\ . \\ \left(u(D(\nu))_{.j} + \delta_j^-\right)\lambda_j & \text{if } \lambda_j \le 0 \end{matrix}$$

Equation 3.2 follows immediately from (3.3). ∎

The conjugate functions of $\psi_{D(\nu)}$ can be used to obtain the conjugate function of co $\psi_{D(\nu)}$ and a bound on $\psi^*(u)$.

PROPOSITION 3.2 The conjugate function of ψ is bounded by

$$\psi^*(u) \geq \left(\text{co } \psi_{D(\nu)}\right)^*(u) = \sup_{\nu=1,\ldots,N} \psi_{D(\nu)}^*. \tag{3.4}$$

Proof: From the definition in (3.1) and (2.13)

$$\psi^*(u) = \sup_{t\in\mathbf{R}^m} [ut - \psi(t)]$$

$$= \left(\text{co } \psi_{D(\nu)}\right)^*(u).$$

From Theorem 16.5 in Rockafellar [1970], $\left(\text{co } \psi_{D(\nu)}\right)^* (u) = \sup_{\nu=1,\ldots,N} \psi_{D(\nu)}^*(u)$, proving (3.4). ∎

From (3.4) and (3.2), the lower bounding function on $\psi^*(u)$ is

$$\psi^*(u) \geq \sup_{\nu=1,\ldots,N} \psi_{D(\nu)}^*(u) = \begin{cases} 0 & \text{if } -\delta_j^- \leq u\left(D_{(\nu)}\right)_j \leq \delta_j^+, \\ & \quad j=1,\ldots,m;\ \nu=1,\ldots,N, \\ +\infty & \quad \text{otherwise.} \end{cases} \tag{3.5}$$

The bound in (3.5) can be useful when a dual program is formulated in place of a primal problem involving $\psi(t)$. The optimization for $\psi(t)$ is replaced by additional feasibility conditions in the dual. Note also that (3.5) provides an alternative method for finding $\left(\text{co } \psi_D(\nu)\right)(t)$. We have

$$(\text{co } \psi_{D(\nu)}(t) = \sup_{u\in\mathbf{R}^m} \{ut - \sup_{\nu=1,\ldots,N} \psi_{D(\nu)}^*(u)\}$$

$$= \sup_{u\in\mathbf{R}^m} \{ut \mid -\delta_j^- \leq u\left(D(\nu)\right)_{.j} \leq \delta_j^+,$$

$$j=1,\ldots,m;\ \nu=1,\ldots,N\}. \tag{3.6}$$

For problems with special structure, the dual formulation for co $\psi_{D(\nu)}$ in (3.6) may provide an efficient method for calculating co $\psi_{D(\nu)}$.

The subgradients of ψ_D are other useful properties of ψ_D when used as an approximation in optimization problems involving ψ.

PROPOSITION 3.3. The subdifferential of $\psi_D(t)$ is given by

$$\partial\ \psi_D(t) = \{\pi \mid \pi = \sigma\, D^{-1},\ \sigma_j \in \Sigma^j,\ j=1,\ldots,m\} \tag{3.7}$$

where

$$\Sigma^j = \begin{cases} -\delta_j^- & \text{if } (D^{-1}t)_j < 0 \\ [-\delta_j^-,\ \delta_j^+] & \text{if } (D^{-1}t)_j = 0 \\ \delta_j^+ & \text{if } (D^{-1}t)_j > 0. \end{cases}$$

Proof: From (3.6), for N=1, and letting $t = D\lambda$,

$$\psi_{D(\nu)}(t) = \sup_{u\in \mathbf{R}^m} \{(u\ D)\lambda \mid -\delta_j^- \le u(D(\nu))_{.j} \le \delta_j^+,\ j = 1,\ldots,m\} \tag{3.8}$$

$$= \sum_{j=1}^{m} \begin{cases} -\delta_j^-\, \lambda_j, & \lambda_j \le 0 \\ \delta_j^+\, \lambda_j, & \lambda_j \ge 0, \end{cases}$$

where $\lambda_j = (D^{-1}t)_j$. Note that π is a subgradient of $\psi_{D(\nu)}$ at t if $\pi t = \psi_{D(\nu)}(t)$ and $\pi z \le \psi_{D(\nu)}(z)$ for all z. Hence, any π such that

$$\pi t = \sum_{j=1}^{m} \begin{cases} -\delta_j^-(D^{-1}t)_j\ , & (D^{-1}t)_j \le 0 \\ \delta_j^+\ (D^{-1}t)_j\ , & (D^{-1}t)_j \ge 0, \end{cases} \tag{3.9}$$

and

$$-\ \delta_j^- \le \pi\ (D(\nu))_{.j} \le \delta_j^+,\ j=1,\ldots,m, \tag{3.10}$$

is a subgradient. The conditions in (3.9) and (3.10) determine $\partial\psi_D$ in (3.7). ∎

The subdifferential of co $\psi_{D(\nu)}$ is also useful in approximating optimization problems involving ψ. From (3.6), this is

$$\partial(\mathrm{co}\ \psi_{D(\nu)})(t) = \{\pi \mid \pi \in \operatorname*{argsup}_{u\in\mathbf{R}^m} \{ut \mid -\delta_j^- \le u(D(\nu))_{.j} \le \delta_j^+,\ j = 1,\ldots,m;\ \nu=1,\ldots,N\}. \tag{3.11}$$

As in the calculation of co $\psi_{D(\nu)}$, the expression in (3.11) may be readily calculable when co $\psi_{D(\nu)}$ has special structure.

A subgradient of co $\psi_{D(\nu)}$ is also valuable in calculating a primal solution x* such that $cx^* \simeq \psi(t)$. For $\pi \in \partial$ co $\psi_{D(\nu)}(t)$, let B be a maximal, linearly independent set of columns chosen from $A^* = \{A_{.i} \mid c_i - \pi A_{.i} \leq 0\}$. In general, B has full rank, and we let $x^* = B^{-1}t$. We then have $cx^* \leq$ co $\psi_{D(\nu)}(t)$ for any choice of B from A^*. If $\{D(\nu), \nu = 1, \ldots, N\}$ is sufficiently large, then for all $A_{.i} \in A^*$, $c_i - \pi A_{.i} = 0$, and $cx^* = \psi(t)$. When B does not have full rank, a looser approximation is obtained by sequentially including columns into B in increasing order of $c_i - \pi A_{.i}$.

Determining which bases to include in $\mathcal{D}$ is an important aspect of the evaluation of the sublinear approximation of ψ. From Proposition 2.4, if all dual feasible bases are included in $\mathcal{D}$, then the sublinear approximation is exact. In general, including all such bases would be inefficient. Instead, we can describe a sequential procedure, in which, new bases are added to $\mathcal{D}$ in order to improve the approximation as much as possible.

A suitable beginning basis, such as D(1) = I, is chosen that provides a rough approximation of ψ. Assume that N bases, D(1),...,D(N), have been chosen and that ψ is inf-compact (i.e., has compact level sets, (Wets [1973]). Consider the level sets

$$\mathrm{lev}_\alpha\, \psi :\; = \{t \mid \psi(t) \leq \alpha\}$$

$$\text{and} \quad \mathrm{lev}_\alpha \left(\mathrm{co}\, \psi_{D(\nu)}\right) = \{t \mid \mathrm{co}\, \psi_{D(\nu)}(t) \leq \alpha\}).$$

We would like to make $\mathrm{lev}_\alpha(\mathrm{co}\, \psi_{D(\nu)})$ as close as possible to $\mathrm{lev}_\alpha\, \psi$. To find these level sets, we use the following result.

<u>PROPOSITION 3.4.</u> If f is a polyhedral function such that

$$f(t) := \inf \{\tau \mid \binom{\tau}{t} = \sum_{i=1}^{k} \binom{\tau^i}{t^i} \lambda^i,\ \lambda^i \geq 0,\ i = 1,\ldots,k\},$$

then $$\mathrm{lev}_\alpha f = \{t \mid t = \sum_{i=1}^{K} \left(\frac{\alpha}{\tau^i}\right) t^i \lambda^i, \ \sum_{i=1}^{K} \lambda^i \leq 1, \ \lambda^i \geq 0, \quad i = 1,\ldots, K\} \tag{3.12}$$
$$= \mathrm{co}\ \{\left(\frac{\alpha}{\tau^i}\right)t^i, \ i = 1,\ldots,k\}.$$

Proof: Suppose that $\mu^i t^i \in \mathrm{lev}_\alpha f$, then $\mu^i \tau^i \leq \alpha$, so $\mathrm{lev}_\alpha f$ includes $\left(\frac{\alpha}{\tau^i}\right)t^i\lambda^i$ for $0 \leq \lambda^i \leq 1$ and, by convexity, $\mathrm{co}\{\left(\frac{\alpha}{\tau^i}\right)t^i, \ i=1,\ldots,K\}$. If $\bar{t} \in \mathrm{lev}_\alpha f$, then $\bar{t} = \sum_{i=1}^{K} t^i\mu^i$ where $f(\bar{t}) = \sum_{i=1}^{K} \tau^i\mu^i$. Let $\mu^i = \left(\frac{\alpha}{\tau^i}\right)\bar{\lambda}^i$. Since $f(\bar{t}) \leq \alpha$, $\sum_{i=1}^{K} \bar{\lambda}^i \leq 1$ and $\bar{t}$ belongs to $\mathrm{co}\{\left(\frac{\alpha}{\tau^i}\right)t^i, \ i=1,\ldots,K\}$. ∎

From Proposition 3.4, we have that
$$\mathrm{lev}_1\ \psi = \mathrm{co}\ \{(q_j^{-1}\ W_{.j}), \ j = 1,\ldots,n\}, \tag{3.13}$$
and
$$\mathrm{lev}_1\ (\mathrm{co}\ \psi_{D(\nu)}) = \mathrm{co}\ \{\left(\frac{1}{\delta_j\pm}\right)(\pm(D(\nu))_{.j}, \ j=1,\ldots,m; \ \nu=1,\ldots,N\}. \tag{3.14}$$

To improve the approximation of ψ by co $\psi_{D(\nu)}$, we would like to minimize the (Hausdorff) distance between $\mathrm{lev}_1\ \psi$ and $\mathrm{lev}_1(\mathrm{co}\ \psi_{D(\nu)})$. We try to do this by creating new bases with
$$t^\nu \in \mathrm{argmax}\ \{\mathrm{dist}(t, \ \mathrm{lev}_1(\mathrm{co}\psi_{D(\nu)}) \mid t \in \mathrm{lev}_1\ \psi\}. \tag{3.15}$$
The nonconvex optimization problem in (3.15) can be solved by searching through the extreme points of $\mathrm{lev}_1\ \psi$, but, in practice, search would only continue until a sufficiently large distance is found (according to some user-specified criterion). When t^ν is found, new matrices, $D(N+1)\ldots,D(N+q)$, are created so that

$$\mathrm{pos}\ [t^\nu, \ D(\nu),-D(\nu), \ \nu=1,\ldots,N] = \bigcup_{\nu=1}^{N+q} \mathrm{pos}[D(\nu),-D(\nu)]. \tag{3.16}$$

Another possibility for finding new matrices for $\mathcal{D}$ is to choose the $D(\nu)$ as rotations of $D(1)=I$. This procedure provides a systematic way for every region of t to be investigated. A third alternative is to choose (randomly or according to a pattern) various values of t and let $D(\nu)$ be the basis that is optimal in finding $\psi(t)$. All of these procedures require further examination and testing before their relative merits can be established.

Given a collection of chosen bases $\mathcal{D}$, the calculations for a sublinear function approximation can be substantially performed in parallel (Wets [1985]). The steps of the method to find inf $\psi_{D(\nu)}(t)$ (assuming δ_j^+, δ_j^- known) are

1. Perform mN inner products $(D(\nu)^{-1})_j\, t$ and determine $\psi_{D(\nu)}^{j}(t)$ in (2.12).
2. Perform N sums of m elements to find $\psi_{D(\nu)}(t)$ as in (2.11).
3. Sort $\psi_{D(\nu)}(t)$, $\nu=1,\dots,N$, to find the least element.

With mN parallel processors, Steps 1 and 2 above can be performed in $O(m)$ time. With N^2 parallel processors, Step 3 can be performed in $O(\log N)$ time.

<u>PROPOSITION</u> <u>3.5</u>. The running time of the sublinear function method to find inf $\psi_{D(\nu)}(t)$ is $O(m + \log N)$ on max (mN, N^2) parallel processors.

The time can vary from the result in Proposition 3.5, especially if fewer than mN or N^2 processors are available. Taking advantage of sparsity, however, and the availability of more processors (to accelerate the inner product) could decrease the running time.

The result in Proposition 3.5 demonstrates the potential for parallel processing with the sublinear function method given here. Approximate solutions of linear programs can be obtained in linear or better than linear time. This can be compared to parallel processing applied to other linear programming algorithms (Pan and Reif [1985]). From Pan and Reif, if

Karmarkar's projective algorithm [1984] requires $O(m^{1/2})$ iterations in practice (as some conjecture), then its running time is $O(m^{1.5} \log m)$ with m^2 parallel processors. If the simplex method (see Dantzig [1963] for a discussion of average case behavior) requires $O(m)$ iterations in practice, then its running time is $O(m^2)$ on m parallel processors. The sublinear function method compares favorably with either of these general-purpose solution methods.

References

Al-Idrisi, M.M., [1981], "Unconstrained Minimization Algorithms for Functions with Singular or Ill-conditioned Hessian", Ph.D. Dissertation, The University of Michigan, Ann Arbor.

Birge, J.R. and R. J-B. Wets, [1986a], "Designing Approximation Schemes for Stochastic Optimization Problems, in Particular, for Stochastic Programs with Recourse," *Mathematical Programming Study* 27, 54-102.

Birge, J.R. and R.J-B. Wets, [1986b], "A Sublinear Approximation Method for Stochastic Programming."

Cuong, H.T., [1980], "Investigation of Methods for Adaptive Control of Surface Ships," Ph.D. Dissertation, The University of Michigan, Ann Arbor.

Dantzig, G.B. [1963], *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey.

Karmarkar, N.K. [1984], "A New Polynomial Time Algorithm for Linear Programming," *Combinatorica* 4, 373-395.

Pan, V. and J. Reif [1985], "Fast and Efficient Algorithms for Linear Programming and for the Linear Least Squares Problem," Computer Science Department, State University of new York, Albany.

Rockafellar, T.R. [1970], *Convex Analysis*, Princeton University Press, Princeton, New Jersey.

Wets, R. J-B. [1973], "On Inf-Compact Mathematical Programs, *Springer-Verlag Lecture Notes in Computer Science* 3, 426-436.

Wets, R. J-B. [1985], "On Parallel Processors Design for Solving Stochastic Programs," Proceedings of the 6-th Mathematical Programming Symposium, Japanese Mathematical Programming Society, Tokyo, 1985, pp. 13-36; also, IIASA Working Paper WP-85-67, International Institute for Applied Systems Analysis, Laxenburg, Austria.