PIECEWISE-LINEAR APPROXIMATION METHODS FOR NONSEPARABLE CONVEX OPTIMIZATION

by

B. Feijoo
and
R. R. Meyer

Computer Sciences Technical Report #521

December 1984

PIECEWISE-LINEAR APPROXIMATION METHODS FOR NONSEPARABLE CONVEX OPTIMIZATION

B. Feijoo and R. R. Meyer†

Abstract

An algorithm is described for the solution of non-separable convex optimization problems. This method utilizes iterative piecewise-linear approximation of the non-separable objective function, but avoids the curse of dimensionality commonly associated with grid methods for multi-dimensional problems. It is, in fact, a practical approach for linearly-constrained large-scale optimization problems, since the direction finding subproblems reduce to linear problems. The method is particularly appropriate for nonlinear networks, since it preserves the network structure of the constraints. A global convergence proof is given under the assumptions that the objective function is Lipschitz continuous and differentiable and that the feasible set is convex and compact.

† This research was supported in part by NSF Grant MCS8200632.

1. Introduction

The problems to be considered are convex optimization problems of the form:

$$\begin{aligned} &\min \ f(\mathbf{x}) \\ &s.t.\ \mathbf{x} \in F \end{aligned} \tag{P}$$

where f is a convex function on the compact convex set $F \subseteq \mathbf{R}^n$. We further assume that a hyper-rectangle $H = \{\mathbf{x} \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}$ is given such that $F \subseteq H$ and f is *Lipschitz continuous* (i.e., $\forall\ \mathbf{x}, \mathbf{y} \in H,\ |f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|$) and *differentiable* on H.

The algorithm to be described is an iterative *descent* method based on a procedure $\mathcal{P}$ that for a given non-optimal point $\mathbf{x} \in F$, returns a set $\mathcal{P}(\mathbf{x}) \subseteq F$ with the property that $\forall\ \mathbf{y} \in \mathcal{P}(\mathbf{x})$, $\mathbf{d} = \mathbf{y} - \mathbf{x}$ is a descent direction for f at $\mathbf{x}$. A line search is then performed along the direction $\mathbf{d}$ and a new iterate found. More precisely, for a feasible point $\mathbf{x}$ and a descent direction $\mathbf{d}$ we consider the problem:

$$\begin{aligned} &\min_{\theta} \ f(\mathbf{x} + \theta \mathbf{d}) \\ &s.t.\ \mathbf{x} + \theta \mathbf{d} \in F \\ &\qquad \theta \geq 0, \end{aligned} \tag{S}$$

and define a search map $\mathcal{S}$ by $\mathcal{S}(\mathbf{x}, \mathbf{d}) = \{\mathbf{y} \mid \mathbf{y} = \mathbf{x} + \theta^* \mathbf{d}$, and θ^* solves (S) $\}$. For simplicity we initially assume exact line searches. The convergence proof is easily extended to any of the inexact line searches based on Armijo-Goldstein rules (see, e.g., [Luenberger, 84]). A proof for a modified Goldstein search is given in the Appendix.

In this general setting the algorithm is defined as follows:

Algorithm 1.1:

Step 0: *Let* $\mathbf{x}^1$ *be a starting feasible solution and* $j \leftarrow 1$.

Step 1: *If* $\mathbf{x}^j$ *satisfies the stopping criterion then stop,*

else

compute $\tilde{\mathbf{x}}^j \in \mathcal{P}(\mathbf{x}^j)$.

Step 2: *Let* $\mathbf{d}^j := \tilde{\mathbf{x}}^j - \mathbf{x}^j$. *Obtain* $\mathbf{x}^{j+1} \in \mathcal{S}(\mathbf{x}^j, \mathbf{d}^j)$. *Set* $j \leftarrow j + 1$ *and go to Step 1.*

We define the *shifted function* $f_{\mathbf{z}}(\mathbf{x}) := f(\mathbf{x}) - f(\mathbf{z})$. Our procedure $\mathcal{P}$ involves approximating $f_{\mathbf{x}^j}$ by a separable function $s_{\mathbf{x}^j}$ that will be called the *underlying separable approximation* to $f_{\mathbf{x}^j}$ centered at $\mathbf{x}^j$. A method of separable optimization is then used to obtain $\tilde{\mathbf{x}}^j$ with the desired descent property. In the case in which the feasible region F is given by a set of network constraints and the separable approximation is piecewise-linear, then the approximating problem can be transformed into a linear network problem that can be solved using the very fast algorithms available for such problems.

Note that any other procedure $\mathcal{P}$ that produces a descent direction and defines a closed map will give rise to a convergent algorithm (see section 3). For example, it's well known that optimization of the linearized objective function will produce a descent direction provided that $\mathbf{x}^j$ is feasible and non-optimal, and this procedure is closed. In the linearly constrained case this is the well-known Frank-Wolfe method.

In section 2 we will describe in detail the nature of the underlying separable approximations. Section 3 contains a discussion of the piecewise-linear approximation method used to attack the separable subproblems in order to obtain the desired descent properties, and a convergence proof for algorithm 1.1 under the assumption that the stopping criterion is optimality of the current iterate. Section 4 includes computational considerations with emphasis on 1) generating only the needed segments of the piecewise-linear approximation and 2) the linear programming or network subproblems that result when F has appropriate structure.

2. Underlying separable approximations

Let $\mathbf{z}$ be a feasible solution of the problem (P). We define the *underlying separable approximation* to $f_{\mathbf{z}}$ as follows:

$$s_{\mathbf{z}}(\mathbf{x}) := \sum_{i=1}^{n} s_{\mathbf{z}}^{(i)}(x_i)$$

where $s_{\mathbf{z}}^{(i)}(x_i) = f_{\mathbf{z}}(\mathbf{z} + (x_i - z_i)\mathbf{e}^i)$, i.e., all variables except the i-th (which has value x_i) are fixed at their values at the point $\mathbf{z}$, and $\mathbf{e}^i$ is the i-th canonical unit vector.

Our first lemma establishes some useful properties of the function $s_{\mathbf{z}}$.

Lemma 2.1. *$s_{\mathbf{z}}$ satisfies the following properties:*

(1) *$s_{\mathbf{z}}$ is a separable, Lipschitz continuous, differentiable convex function on H;*

(2) $s_{\mathbf{z}}(\mathbf{z}) = 0$;

(3) $\nabla s_{\mathbf{z}}(\mathbf{z}) = \nabla f_{\mathbf{z}}(\mathbf{z}) = \nabla f(\mathbf{z})$.

Proof: The Lipschitz continuity, differentiability and convexity of $s_{\mathbf{z}}$ are inherited from those of $f_{\mathbf{z}}$. Property (2) is easily checked from the definition of $s_{\mathbf{z}}$. Property (3) is also straightforward: Let $1 \leq j \leq n$, then

$$\frac{\partial s_{\mathbf{z}}}{\partial x_j}(\mathbf{x}) = \frac{ds_{\mathbf{z}}^{(j)}}{dx_j}(x_j) = \nabla f_{\mathbf{z}}(\mathbf{z} + (x_j - z_j)\mathbf{e}^j) \cdot \mathbf{e}^j = \frac{\partial f_{\mathbf{z}}}{\partial x_j}(\mathbf{z} + (x_j - z_j)\mathbf{e}^j)$$

hence, letting $\mathbf{x} = \mathbf{z}$ we get

$$\frac{\partial s_{\mathbf{z}}}{\partial x_j}(\mathbf{z}) = \frac{\partial f_{\mathbf{z}}}{\partial x_j}(\mathbf{z})$$

and this concludes the proof. ∎

The concepts of feasible direction and descent direction are needed for the discussion of the following results. We now give a formal definition of these terms.

Definition 2.2. *Given $\mathbf{x} \in F$, we will say $\mathbf{d}$ is a feasible direction at $\mathbf{x}$ if there is an $\bar{\alpha} > 0$ such that $\mathbf{x} + \alpha \mathbf{d} \in F$ for all $\alpha \in [0, \bar{\alpha}]$. Given $\mathbf{x} \in F$ and f we say that $\mathbf{d}$ is a descent direction for f at $\mathbf{x}$ if there is an $\hat{\alpha} > 0$ such that $f(\mathbf{x} + \alpha \mathbf{d}) < f(\mathbf{x})$ for all $\alpha \in [0, \hat{\alpha}]$.*

It's well known that if f is convex and differentiable, $\mathbf{d}$ is a descent direction for f at $\mathbf{x}$ if and only if $\nabla f(\mathbf{x}) \cdot \mathbf{d} < 0$.

The following lemma establishes the relationship between the optimal solution $\mathbf{x}^*$ of (P) and that of the underlying separable problem centered at $\mathbf{x}^*$.

Lemma 2.3. *A feasible solution $\mathbf{x}^*$ is an optimal solution of the separable problem:*

$$\begin{aligned} \min \quad & s_{\mathbf{x}^*}(\mathbf{x}) \\ s.t. \quad & \mathbf{x} \in F \end{aligned} \qquad (\mathrm{SP}(\mathbf{x}^*))$$

if and only if $\mathbf{x}^*$ is an optimal solution of the original problem (P).

Proof:

[⇒]

Let $\mathbf{d}$ be a feasible direction at $\mathbf{x}^*$, (if $\mathbf{d}$ doesn't exist, then $F = \{\mathbf{x}^*\}$ and the lemma is trivial). Since $\mathbf{x}^*$ is optimal for $(\mathrm{SP}(\mathbf{x}^*))$ we have $\nabla s_{\mathbf{x}^*}(\mathbf{x}^*) \cdot \mathbf{d} \geq 0$. But by lemma 2.1(3) we have $\nabla s_{\mathbf{x}^*}(\mathbf{x}^*) = \nabla f(\mathbf{x}^*)$ so $\nabla f(\mathbf{x}^*) \cdot \mathbf{d} \geq 0$ also, and by the convexity of f, $f(\mathbf{x}^* + \alpha \mathbf{d}) \geq f(\mathbf{x}^*)$, $\forall \alpha \geq 0$. This is true for any feasible direction at $\mathbf{x}^*$, hence $\mathbf{x}^*$ is an optimal solution of (P).

[⇐]

Follows by reversing the roles of $s_{\mathbf{x}^*}$ and f. ∎

The next lemma establishes a key descent relationship between f and its separable approximations. It shows that it suffices to find a feasible point that improves the separable approximation in order to get a descent direction for the original objective function.

Lemma 2.4. *Let* $\mathbf{y}$, $\mathbf{z} \in F$, *if* $s_{\mathbf{z}}(\mathbf{y}) < 0$, *then* $\mathbf{d} = \mathbf{y} - \mathbf{z}$ *is a descent direction for both* $s_{\mathbf{z}}$ *and* f *at* $\mathbf{z}$.

Proof: Since $s_{\mathbf{z}}$ is convex and differentiable and $s_{\mathbf{z}}(\mathbf{z}) = 0$, then for every $\mathbf{x} \in F$ we have

$$\nabla s_{\mathbf{z}}(\mathbf{z}) \cdot (\mathbf{x} - \mathbf{z}) \leq s_{\mathbf{z}}(\mathbf{x})$$

thus

$$\nabla s_{\mathbf{z}}(\mathbf{z}) \cdot (\mathbf{y} - \mathbf{z}) \leq s_{\mathbf{z}}(\mathbf{y}) < 0$$

from lemma 2.1(3) we then have

$$\nabla f(\mathbf{z}) \cdot (\mathbf{y} - \mathbf{z}) < 0$$

so $\mathbf{d}$ is a descent direction for both $s_{\mathbf{z}}$ and f at $\mathbf{z}$. ∎

3. Piecewise-linear approximations and convergence

Since the problem (SP($\mathbf{x}$)) is still a nonlinear programming problem, we consider now an algorithm based on the solution of a piecewise-linear direction-finding problem at each iteration. In this section we describe in detail the piecewise-linear approximating functions to be used in our procedure $\mathcal{P}$ and prove that, for this choice of $\mathcal{P}$, algorithm 1.1 produces a sequence of iterates converging to an optimal solution of (P).

Given a feasible point $\mathbf{z}$ of (P) and a vector $\Lambda = (\lambda_1, \ldots, \lambda_n) > \mathbf{0}$ of gridsizes, a piecewise-linear approximation $\tilde{f}$ to $s_{\mathbf{z}}$ is defined as follows:

$$\tilde{f}(\mathbf{z}, \Lambda, \mathbf{x}) = \sum_{i=1}^{n} \tilde{f}_i(\mathbf{z}, \Lambda, x_i)$$

where

$$\tilde{f}_i(\mathbf{z}, \Lambda, x_i) = \begin{cases} s_{\mathbf{z}}^{(i)}(z_i + (k_i - 1)\lambda_i) \quad + \quad c_i^{+k_i}\{x_i - (z_i + (k_i - 1)\lambda_i)\} \\ \qquad\qquad \text{for } (k_i - 1)\lambda_i \leq x_i - z_i \leq k_i\lambda_i \\ \qquad\qquad k_i = 1, 2, \ldots, s_i \\ \\ s_{\mathbf{z}}^{(i)}(z_i - (k_i - 1)\lambda_i) \quad + \quad c_i^{-k_i}\{x_i - (z_i - (k_i - 1)\lambda_i)\} \\ \qquad\qquad \text{for } -k_i\lambda_i \leq x_i - z_i \leq -(k_i - 1)\lambda_i \\ \qquad\qquad k_i = 1, 2, \ldots, s_i' \end{cases}$$

and

$$c_i^{+k_i} = \frac{\{s_{\mathbf{z}}^i(z_i + k_i\lambda_i) - s_{\mathbf{z}}^i(z_i + (k_i - 1)\lambda_i)\}}{\lambda_i}$$

$$-c_i^{-k_i} = \frac{\{s_{\mathbf{z}}^i(z_i - k_i\lambda_i) - s_{\mathbf{z}}^i(z_i - (k_i - 1)\lambda_i)\}}{\lambda_i} \quad .$$

and s_i and s'_i are chosen so that $z_i + s_i\lambda_i = u_i$ and $z_i - s'_i\lambda_i = l_i$ (For notational simplicity, we assume that at any iteration all segments used to construct an approximation $\tilde{f}_i$ are *equal*; in practice the segments near the boundaries defined by l_i and u_i are generally smaller than λ_i, but this poses no problems in utilizing the theory to be developed since the proofs below merely utilize the fact that λ_i is an *upper bound* on segment size. Moreover, the segments of $\tilde{f}_i$ may be generated *as needed* starting at z_i, and it is seldom necessary to generate more than a few such segments. In a sense, $\tilde{f}_i$ is implicitly rather than explicitly generated).

Even though the formulation looks complicated, the idea is simply to approximate each of the $s^i_{\mathbf{z}}$ by the standard piecewise-linear approximation centered at z_i, with gridsize λ_i. It follows from the convexity of $s^{(i)}_{\mathbf{z}}$ that $s^{(i)}_{\mathbf{z}}(x_i) \leq \tilde{f}_i(\mathbf{z}, \Lambda, x_i)$ for all $x_i \in [l_i, u_i]$. This property is used in the convergence proof below.

Given the iterate $\mathbf{x}^j$ constructed in iteration j, we define an approximating problem

$$\min_{\mathbf{x}} \sum_{i=1}^{n} \tilde{f}_i(\mathbf{x}^j, \Lambda^j, x_i) \qquad (\mathrm{AP}(\mathbf{x}^j, \Lambda^j))$$
$$s.t.\ \mathbf{x} \in F$$

that will be the basis for our procedure $\mathcal{P}$ for finding the search direction.

Procedure $\mathcal{P}(\mathbf{x}^j)$**:**

Step 1: *If* $j = 1$, *then choose* Λ^1 *such that* $\mathbf{0} \leq \Lambda^1 \leq \mathbf{u} - \mathbf{l}$,

else

for $\alpha \in (0,1)$ *let* $\Lambda^j \leftarrow \alpha\Lambda^{j-1}$.

Step 2: *Obtain an optimal solution* $\tilde{\mathbf{x}}^j$ *of* $(\mathrm{AP}(\mathbf{x}^j, \Lambda^j))$.

If $\tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) = 0$, *then* $\Lambda^j \leftarrow \alpha\Lambda^j$ *and repeat Step 2,*

else

stop, returning $\{\tilde{\mathbf{x}}^j\}$.

It should be noted that $\tilde{f}$ is defined in such a way that $\tilde{f}(\mathbf{z}, \Lambda, \mathbf{z}) = 0$ and thus the test in Step 2 of $\mathcal{P}$ determines if the mesh is fine enough to produce a descent direction for $\tilde{f}$ (and hence for $s_{\mathbf{z}}$ and f). In each call to $\mathcal{P}$ the mesh size is reduced by a factor of α. If this reduction is not enough to produce an improvement in the piecewise-linear approximation we repeatedly cut down the mesh size until such an improvement is achieved. We know from earlier results for piecewise-linear approximations of separable functions (see, e.g., [Kamesam and Meyer, 82]) that if $\mathbf{x}^j$ is not an optimal solution for $(\mathrm{SP}(\mathbf{x}^j))$, then for a small enough Λ^j a better feasible solution will be found, i.e., $\tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) < 0$. In most cases in practice, however, only the initial optimal solution is needed. It is shown in theorem 3.5 below that an improved solution of the approximating problem yields a descent direction for f and consequently the objective values of the iterates form a strictly decreasing sequence.

The next three lemmas establish the uniform convergence of the approximating functions considered.

Lemma 3.1. *If K is an index set such that $\mathbf{x}^j \xrightarrow{j\in K} \bar{\mathbf{x}}$, then $s_{\mathbf{x}^j} \xrightarrow{j\in K} s_{\bar{\mathbf{x}}}$ uniformly.*

Proof: See Appendix.

Lemma 3.2. *If $\Lambda^j \xrightarrow{j\in K} \mathbf{0}$ for $K \subseteq \mathbf{N}$. Then $\tilde{f}(\mathbf{z},\Lambda^j,\mathbf{x}) \xrightarrow{j\in K} s_{\mathbf{z}}(\mathbf{x})$ uniformly in $\mathbf{x}$ and $\mathbf{z}$.*

Proof: Let $j \in K$, $\mathbf{z} \in F$ and $\mathbf{x} \in F$. Then if we define $E_i = |\tilde{f}_i(\mathbf{z},\Lambda^j,x_i) - s_{\mathbf{z}}^{(i)}(x_i)|$, we have

$$|\tilde{f}(\mathbf{z},\Lambda^j,\mathbf{x}) - s_{\mathbf{z}}(\mathbf{x})| \leq \sum_{i=1}^{n} E_i$$

An estimate for E_i in the presence of a Lipschitz condition for $s_{\mathbf{z}}^{(i)}$ is well known (see, e.g., [Thakur, 78]):

$$E_i \leq \frac{L\lambda_i^j}{2}$$

So now we have that

$$|\tilde{f}(\mathbf{z},\Lambda^j,\mathbf{x}) - s_{\mathbf{z}}(\mathbf{x})| \leq \frac{L}{2}\sum_{i=1}^{n}\lambda_i^j \leq \frac{L}{2}\|\Lambda^j\|_1$$

and since we have $\Lambda^j \xrightarrow{j\in K} \mathbf{0}$ the convergence is uniform. ∎

Lemma 3.3. *If $\Lambda^j \xrightarrow{j\in K} \mathbf{0}$ and $\mathbf{x}^{j+1} \xrightarrow{j\in K} \bar{\mathbf{x}}$ for $K \subseteq \mathbf{N}$ then $\tilde{f}(\mathbf{x}^{j+1},\Lambda^j,\cdot) \xrightarrow{j\in K} s_{\bar{\mathbf{x}}}$ uniformly.*

Proof: Since

$$|\tilde{f}(\mathbf{x}^{j+1},\Lambda^j,\mathbf{x}) - s_{\bar{\mathbf{x}}}(\mathbf{x})| \leq |\tilde{f}(\mathbf{x}^{j+1},\Lambda^j,\mathbf{x}) - s_{\mathbf{x}^{j+1}}(\mathbf{x})| + |s_{\mathbf{x}^{j+1}}(\mathbf{x}) - s_{\bar{\mathbf{x}}}(\mathbf{x})|$$

the result follows from lemmas 3.1 and 3.2. ∎

The following theorem uses the continuity properties of the approximating functions to show that the optimal solutions of the approximating problems have the appropriate continuity property.

Theorem 3.4. *Let* $\mathbf{x}^*$, $\tilde{\mathbf{x}} \in F$ *and* $K \subseteq \mathbf{N}$ *be an index set such that*

(1) $\mathbf{x}^j \xrightarrow{j \in K} \mathbf{x}^*$.

(2) $\tilde{\mathbf{x}}^j$ *is an optimal solution of* $(\mathrm{AP}(\mathbf{x}^j, \Lambda^j))$, *for* $j \in K$ *with* $\tilde{\mathbf{x}}^j \xrightarrow{j \in K} \tilde{\mathbf{x}}$.

(3) $\Lambda^j \xrightarrow{j \in K} \mathbf{0}$.

Then $\tilde{\mathbf{x}}$ *is an optimal solution of* $(\mathrm{SP}(\mathbf{x}^*))$.

Proof: Let $\mathbf{x}^{**}$ be an optimal solution of $(\mathrm{SP}(\mathbf{x}^*))$ and choose $\epsilon > 0$. By lemma 3.3 and the continuity of $s_{\mathbf{x}^*}$, $\exists\, N$ such that for $j \in K$, $j \geq N$ we have $|\tilde{f}(\mathbf{x}^j, \Lambda^j, \mathbf{x}) - s_{\mathbf{x}^*}(\mathbf{x})| < \epsilon$, $\forall\, \mathbf{x} \in H$ and $|s_{\mathbf{x}^*}(\tilde{\mathbf{x}}^j) - s_{\mathbf{x}^*}(\tilde{\mathbf{x}})| < \epsilon$. In particular

$$\tilde{f}(\mathbf{x}^j, \Lambda^j, \mathbf{x}^{**}) < s_{\mathbf{x}^*}(\mathbf{x}^{**}) + \epsilon \tag{1}$$

$$s_{\mathbf{x}^*}(\tilde{\mathbf{x}}) < s_{\mathbf{x}^*}(\tilde{\mathbf{x}}^j) + \epsilon \tag{2}$$

and similarly

$$s_{\mathbf{x}^*}(\tilde{\mathbf{x}}^j) < \tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) + \epsilon \tag{3}$$

but from (2) and (3) we get

$$s_{\mathbf{x}^*}(\tilde{\mathbf{x}}) < \tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) + 2\epsilon \tag{4}$$

On the other hand, since $\mathbf{x}^{**} \in F$ and $\tilde{\mathbf{x}}^j$ is optimal for $(\mathrm{AP}(\mathbf{x}^j, \Lambda^j))$, we have

$$\tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) \leq \tilde{f}(\mathbf{x}^j, \Lambda^j, \mathbf{x}^{**}) \tag{5}$$

so combining inequalities (1), (4) and (5) yields

$$s_{\mathbf{x}^*}(\tilde{\mathbf{x}}) < s_{\mathbf{x}^*}(\mathbf{x}^{**}) + 3\epsilon$$

Since ϵ is arbitrary, $s_{\mathbf{x}^*}(\tilde{\mathbf{x}}) \leq s_{\mathbf{x}^*}(\mathbf{x}^{**})$ and the optimality of $\tilde{\mathbf{x}}$ is established. ∎

Theorem 3.5. *In algorithm 1.1 with* $\mathcal{P}$ *defined as above, if* $\mathbf{x}^j$ *is not an optimal solution of the original problem (P), then* $f(\mathbf{x}^{j+1}) < f(\mathbf{x}^j)$.

Proof: Since $\mathbf{x}^j$ is not optimal for problem (P) we have, by lemma 2.3, that $\mathbf{x}^j$ is *not* optimal for $(\mathrm{SP}(\mathbf{x}^j))$. As mentioned above, we know that $\mathcal{P}$ will produce $\tilde{\mathbf{x}}^j$ such that $\tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) < 0$. The approximation $\tilde{f}$ satisfies $\tilde{f}(\mathbf{x}^j, \Lambda^j, \mathbf{x}) \geq s_{\mathbf{x}^j}(\mathbf{x}),\ \ \forall\, \mathbf{x} \in H$. These inequalities imply

$$s_{\mathbf{x}^j}(\tilde{\mathbf{x}}^j) \leq \tilde{f}(\mathbf{x}^j, \Lambda^j, \tilde{\mathbf{x}}^j) < 0$$

so by lemma 2.4 $\mathbf{d}^j$ is a *descent* direction for f at $\mathbf{x}^j$, and since $\mathbf{x}^{j+1} \in \mathcal{S}(\mathbf{x}^j, \mathbf{d}^j)$ we have the desired result. ∎

From the preceding theorem it follows that $\{f(\mathbf{x}^j)\}$ converges. We now show that the limit of this sequence is the optimal value of (P) (we will assume that a full sequence $\{\mathbf{x}^j\}$ is generated, otherwise the method terminates at an optimal solution).

Theorem 3.6. *The iterates $\mathbf{x}^j$ generated by algorithm 1.1, with $\mathcal{P}$ defined as above, have the property that $\{f(\mathbf{x}^j)\}_{j=1}^{\infty}$ converges to the optimal value of the original problem (P).*

Proof: Let $\{\mathbf{x}^j\}_{j=1}^{\infty}$ be a sequence of iterates generated by algorithm 1.1, and let $\{\tilde{\mathbf{x}}^j\}_{j=1}^{\infty}$ be the sequence produced by step 1 of the algorithm. The sequences $\{\mathbf{x}^j\}_{j=1}^{\infty}$, $\{\mathbf{x}^{j+1}\}_{j=1}^{\infty}$ and $\{\tilde{\mathbf{x}}^j\}_{j=1}^{\infty}$ are contained in F, a compact set, so they have accumulation points in F, say $\mathbf{x}^*$, $\hat{\mathbf{x}}$ and $\tilde{\mathbf{x}}$ respectively. Without loss of generality there exists a set $K \subseteq \mathbf{N}$, such that

$$\mathbf{x}^j \xrightarrow{j\in K} \mathbf{x}^*$$
$$\mathbf{x}^{j+1} \xrightarrow{j\in K} \hat{\mathbf{x}}$$
$$\tilde{\mathbf{x}}^j \xrightarrow{j\in K} \tilde{\mathbf{x}}$$

We want to prove that $\mathbf{x}^*$ is an optimal solution of (P). Assuming $\mathbf{x}^*$ is not an optimal solution of (P) we will get a contradiction. By lemma 2.3 $\mathbf{x}^*$ is *not* an optimal solution of $(\mathrm{SP}(\mathbf{x}^*))$. Since $\Lambda^j \xrightarrow{j\in K} \mathbf{0}$, by theorem 3.4 we have that $\tilde{\mathbf{x}}$ is an optimal solution of $(\mathrm{SP}(\mathbf{x}^*))$, hence $s_{\mathbf{x}^*}(\tilde{\mathbf{x}}) < 0$. By lemma 2.4 $\mathbf{d}^* := \tilde{\mathbf{x}} - \mathbf{x}^*$ is a descent direction for f at $\mathbf{x}^*$.

Now consider the problem

$$\begin{aligned} \min_{\theta} \;\; & f(\mathbf{x}^* + \theta\mathbf{d}^*) \\ s.t. \;\; & \mathbf{x}^* + \theta\mathbf{d}^* \in F \\ & 0 \le \theta \le 1 \end{aligned} \tag{S$'$}$$

and let θ^* be an optimal solution of (S$'$). Defining $\mathbf{x}_s := \mathbf{x}^* + \theta^*\mathbf{d}^*$ we have $f(\mathbf{x}_s) < f(\mathbf{x}^*)$. Now let $\mathbf{y}^{j+1} = \mathbf{x}^j + \theta^*\mathbf{d}^j$ for $\mathbf{d}^j$ as defined in algorithm 1.1. Since $0 < \theta^* \le 1$ we have $\mathbf{y}^j \in F$ and $\mathbf{y}^j \xrightarrow{j\in K} \mathbf{x}_s$. Moreover $f(\mathbf{x}^{j+1}) \le f(\mathbf{y}^j)$ because $\mathbf{x}^{j+1}$ minimizes f over the set $\{\mathbf{x} \mid \mathbf{x} = \mathbf{x}^j + \theta\mathbf{d}^j,\ \mathbf{x} \in F,\ \theta \ge 0\}$. In the limit we then have $f(\hat{\mathbf{x}}) \le f(\mathbf{x}_s) < f(\mathbf{x}^*)$, contradicting $f(\hat{\mathbf{x}}) = f(\mathbf{x}^*)$.

So $\mathbf{x}^*$ is an optimal solution of (P). ∎

Note that the proof of the theorem goes through if $\mathcal{P}$ is any closed mapping with the property that if $\mathbf{x}$ is not optimal for (P) and $\mathbf{y} \in \mathcal{P}(\mathbf{x})$, then $\mathbf{y} - \mathbf{x}$ is a descent direction for f at $\mathbf{x}$.

In the case of problems having network constraints and upper and lower bounds only, there are other algorithms for separable functions ([Rockafellar, 84]) that produce a descent direction and can be used in place of our piecewise-linear approach. Some research has to be done, however, to establish the closedness of these algorithms.

4. Computational Implementation

The replacement of the nonseparable objective function by a piecewise-linear approximating function is primarily useful in the case in which the feasible set F is given by a set of *linear constraints*. In this instance, the approximating problems may be solved as ordinary linear programs by the well-known techniques of separable programming [Bazaraa and Shetty, 79]. At any iteration, in fact, the problem to be solved may be thought of as the problem resulting from the application of a separable programming approach to a separable function of the form $s_{\mathbf{z}}(\mathbf{x})$. In the non-separable case the function being approximated thus changes at each iteration. It is well-known that the piecewise-linear approximating problems $\mathrm{AP}(\mathbf{x}^j, \Lambda^j)$ may be solved by generating the segments of $\tilde{f}_i$ "as needed" and that an optimal solution of such an approximating problem may generally be determined and verified on the basis of a small number of segments computed in a neighborhood of $\mathbf{x}^j$ (see [Meyer, 83], [Kamesam and Meyer, 82]).

Specializing a bit further, it should be noted that when F is defined by network constraints and the variables x_i correspond to arc flows, the approximating problems are equivalent to ordinary linear network problems (see, e.g., [Jensen and Barnes, 1980], [Kao and Meyer, 1981]). Of particular interest are *large-scale multi-commodity traffic equilibrium problems* (see. e.g., [Pang and Yu, 1982] and [Gavish and Hantler, 82]) arising from urban traffic as well as computer networks. When the coupling between commodities takes place only in the objective function, then the replacement of the non-separable function by a separable functions allows the problem to be decomposed into a set of single commodity networks, which in turn may be solved by the very fast techniques available for such problems (see, e.g., [Glover, et al, 74] and [Grigoriadis and Hsu, 79]). Such an approach also allows the individual single-commodity problems to be solved in parallel on parallel computer architectures such as the CRYSTAL multicomputer [DeWitt, et al, 84]. Details of the CRYSTAL implementation are described in [Feijoo and Meyer 84].

Although an arbitrary feasible starting point may be used in algorithm 1.1, in practice in the absence of knowledge of a good initial point, $\mathbf{x}^1$ is generated by solving a problem of the form $\mathrm{AP}(\bar{\mathbf{x}}, \lambda^0)$, where $\bar{\mathbf{x}}$ is an arbitrary element of H and λ_i^0 is typically $\frac{(u_i - l_i)}{4}$.

Thus far, three problems have been tested. All three are related to urban traffic equilibrium problems. The sources of these problems are: Problem 1, [Pang, 83]; Problem 2, [Bertsekas and Gafni, 82]; and Problem 3, [Steenbrink, 74]. The nonseparable objective functions for these problems are as follows:

Problem 1 :
$$f = \sum_k a_k f_k^5 + b_k f_k$$

	Arcs	Nodes	Cmdts.	Cnstr.	Vars.
Problem 1	28	20	2	39	56
Problem 2	12	12	5	60	64
Problem 3	36	9	12	108	443

Table 4.1: Test Problems

	Iter.	Pivots	11/780	CRYSTAL	Obj. value
Problem 1	32	474	8.25 s.	15.2 s.	1010540.6
Problem 2	12	650	6.47 s.	7.0 s.	47.858537
Problem 3	11	2266	47.20 s.	23.5 s.	16957.674

Table 4.2: Computational Results

	Iter.	11/780	Obj. value
Problem 1	32	8.41 s.	1010540.6
Problem 2	6	6.68 s.	47.858538
Problem 3	106	39.83 s.	16957.674

Table 4.3: MINOS results

Problem 2 :
$$f = \sum_k \frac{f_k^3}{3} + \frac{f_k^2}{2} + f_k$$

Problem 3 :
$$f = \sum_k a_k f_k^2 + b_k f_k$$

where f_k represents total flow on link k, i.e., the sum of flows of all commodities through link k.

We ran tests using our algorithm and also MINOS ([Murtagh and Saunders, 83]). Tables 4.1 and 4.2 show the size of the problems and the results obtained using our algorithm. Table 4.3 shows the results obtained using MINOS. All tests, except the ones done on CRYSTAL, were run on a VAX 11/780 using the UNIX Fortran 77 compiler without optimization. The time shown for CRYSTAL is total elapsed time for the algorithm when it is run in parallel on a collection of VAX 11/750 minicomputers, where the number of such computers equals the number of commodities in the problem.

Appendix

In this appendix we will give a modified proof of the main convergence theorem in the case in which an inexact line search, based on Armijo-Goldstein tests, is used. For this case we need to assume that f is continuously differentiable. We also give a proof of lemma 3.1.

First we define a modified Goldstein test so it takes care of the fact we are searching inside a compact set. We denote by $\mathcal{I}_G$ the set of acceptable step lengths.

Modified Goldstein Test. *Let* $\mathbf{x} \in F$ *and let* $\mathbf{d}$ *be a descent direction for* f *at* $\mathbf{x}$. *For a given fixed* ϵ, *with* $0 < \epsilon < \frac{1}{2}$, *define*

$$\mathcal{A}(\mathbf{x},\mathbf{d}) = \{\theta \mid \theta \geq 0,\ \mathbf{x}+\theta\mathbf{d} \in F \ \textit{ and } \ \epsilon \leq \mathcal{Q}(\theta,\mathbf{x},\mathbf{d}) \leq 1-\epsilon\}$$

where

$$\mathcal{Q}(\theta,\mathbf{x},\mathbf{d}) := \frac{f(\mathbf{x}+\theta\mathbf{d}) - f(\mathbf{x})}{\theta\nabla f(\mathbf{x})\cdot\mathbf{d}} .$$

If $\mathcal{A}(\mathbf{x},\mathbf{d}) \neq \emptyset$, *then* $\mathcal{I}_G := \mathcal{A}(\mathbf{x},\mathbf{d})$, *else* $\mathcal{I}_G := \{\theta^*\}$, *where* $\theta^* = \max\{\theta \mid \mathbf{x}+\theta\mathbf{d} \in F\}$. *The corresponding search map is*

$$\mathcal{S}_G(\mathbf{x},\mathbf{d}) = \{\mathbf{y} \mid \mathbf{y} = \mathbf{x}+\theta\mathbf{d},\quad \theta \in \mathcal{I}_G\}$$

If we use this test in algorithm 1.1, let us consider how the proof of theorem 3.6 may be extended. The first part is independent of the line search and thus the conclusion that $\mathbf{d}^* = \tilde{\mathbf{x}} - \mathbf{x}^*$ is a descent direction is still true. It suffices to show that $f(\hat{\mathbf{x}}) < f(\mathbf{x}^*)$.

Let $j \in K$ and $\mathbf{x}^{j+1} = \mathbf{x}^j + \theta^j\mathbf{d}^j$, and suppose $\mathcal{A}(\mathbf{x}^j,\mathbf{d}^j) = \emptyset$. Then $\theta^j \geq 1$ and $\mathcal{Q}(\theta^j,\mathbf{x}^j,\mathbf{d}^j) > 1-\epsilon$, since $\mathcal{Q}(\theta,\mathbf{x}^j,\mathbf{d}^j) \longrightarrow 1$ as $\theta \longrightarrow 0$. Thus, $\mathcal{Q}(\theta^j,\mathbf{x}^j,\mathbf{d}^j) \geq \epsilon$ and

$$f(\mathbf{x}^j+\theta^j\mathbf{d}^j) - f(\mathbf{x}^j) \leq \theta^j\epsilon\nabla f(\mathbf{x}^j)\cdot\mathbf{d}^j \tag{1}$$

On the other hand, if $\mathcal{A}(\mathbf{x}^j,\mathbf{d}^j) \neq \emptyset$ then θ^j clearly satisfies (1). We thus conclude that (1) is satisfied for all $j \in K$. Now since $\mathbf{x}^{j+1} = \mathbf{x}^j + \theta^j\mathbf{d}^j$ we can write

$$\theta^j = \frac{||\mathbf{x}^{j+1} - \mathbf{x}^j||}{||\mathbf{d}^j||}$$

if we define

$$\bar{\theta} := \frac{||\hat{\mathbf{x}} - \mathbf{x}^*||}{||\mathbf{d}^*||}$$

we have $\theta^j \xrightarrow{j\in K} \bar{\theta}$. Taking the limit in (1), we get

$$f(\hat{\mathbf{x}}) \leq f(\mathbf{x}^*) + \epsilon\bar{\theta}\nabla f(\mathbf{x}^*)\cdot\mathbf{d}^* \tag{2}$$

Note that $\bar{\theta} > 0$, since $\theta^j \xrightarrow{j \in K} 0$ implies $\mathcal{Q}(\theta^j, \mathbf{x}^j, \mathbf{d}^j) \xrightarrow{j \in K} 1$, contradicting $\mathcal{Q}(\theta^j, \mathbf{x}^j, \mathbf{d}^j) \leq 1 - \epsilon$. Then from (2), since $\nabla f(\mathbf{x}^*) \cdot \mathbf{d}^* < 0$, we have $f(\hat{\mathbf{x}}) < f(\mathbf{x}^*)$. This concludes the proof. ∎

We now prove lemma 3.1:

Given $\epsilon > 0$, $\exists\ N$ such that if $j \in K$ and $j \geq N$ then $\|\mathbf{x}^j - \bar{\mathbf{x}}\|_1 < \frac{\epsilon}{2nL}$. Then for such ϵ we have

$$
\begin{aligned}
|s_{\mathbf{x}^j}(\mathbf{x}) - s_{\bar{\mathbf{x}}}(\mathbf{x})| &\leq \sum_{i=1}^{n} |s^i_{\mathbf{x}^j}(x_i) - s^i_{\bar{\mathbf{x}}}(x_i)| \\
&= \sum_{i=1}^{n} |f_{\mathbf{x}^j}(\mathbf{x}^j + (x_i - x_i^j)\mathbf{e}^i) - f_{\bar{\mathbf{x}}}(\bar{\mathbf{x}} + (x_i - \bar{x}_i)\mathbf{e}^i)| \\
&\leq \sum_{i=1}^{n} |f(\mathbf{x}^j + (x_i - x_i^j)\mathbf{e}^i) - f(\bar{\mathbf{x}} + (x_i - \bar{x}_i)\mathbf{e}^i)| + |f(\mathbf{x}^j) - f(\bar{\mathbf{x}})| \\
&\leq \sum_{i=1}^{n} L(\|(\mathbf{x}^j + (x_i - x_i^j)\mathbf{e}^i) - (\bar{\mathbf{x}} + (x_i - \bar{x}_i)\mathbf{e}^i)\|_1 + \|\mathbf{x}^j - \bar{\mathbf{x}}\|_1) \\
&\leq \sum_{i=1}^{n} 2L\|\mathbf{x}^j - \bar{\mathbf{x}}\|_1 = 2nL\|\mathbf{x}^j - \bar{\mathbf{x}}\|_1 < \epsilon
\end{aligned}
$$

Since ϵ is arbitrary this concludes the proof. ∎

References

1. M. S. Bazaraa and C. M. Shetty: *Nonlinear Programming, theory and algorithms*, John Wiley and Sons, 1979.

2. D. P. Bertsekas and E. M. Gafni: "Projection methods for minimum cost network flow problems", *Mathematical Programming Study*,17 , 1982, 139-159.

3. D. DeWitt, R. Finkel and M. Solomon: "The CRYSTAL multicomputer: design and implementation experience", Technical report 553, Computer Sciences Department, The University of Wisconsin-Madison, September, 1984.

4. B. Feijoo and R. R. Meyer: "Optimization on the CRYSTAL multicomputer", Technical Report 562, Computer Sciences Department, The University of Wisconsin-Madison, 1984.

5. B. Gavish and S. L. Hantler: "An algorithm for optimal route selection in SNA networks", Research Report RC 9549, IBM T. J. Watson Research Center, Yorktown Heights, N.Y., August, 1982.

6. F. Glover, D. Karney and D. Klingman: "Implementation and computational comparisons of primal, dual, and primal-dual computer codes for minimum cost network flow problems", *Networks*, 4, 191-212, 1974.

7. M. D. Grigoriadis and T. Hsu: "RNET the Rutgers minimum cost network flow subroutine", *SIGMAP Bulletin*, 17-18, 1979.

8. P. A. Jensen and W. J. Barnes: *Network Flow Programming*, John Wiley and Sons, 1980.

9. P. V. Kamesam and R. R. Meyer: "Multipont methods for nonlinear networks", Technical Report 468, Computer Sciences Department, The University of Wisconsin-Madison, 1982.

10. C. Y. Kao and R. R. Meyer: "Secant approximation methods for convex optimization", *Mathematical Programming Study*, 14, 143-162.

11. D. G. Luenberger: *Linear and Nonlinear Programming*, Second Edition, Addison-Wesley, 1984.

12. R. R. Meyer: "Computational aspects of two-segment separable programming", Technical Report 382, *Mathematical Programming*, 26, 21-39, 1983.

13. B. A. Murtagh and M. A. Saunders: "MINOS 5.0 user's guide", Technical Report SOL 83-20, Stanford University, Stanford, 1983.

14. J. S. Pang, private communication, 1983.

15. J. S. Pang and C. S. Yu: "Linearized simplicial decomposition methods for computing traffic equilibria on networks", Technical Report, University of Texas at Dallas, Richardson, Texas, 1982.

16. R. T. Rockafellar, *Network Flows and Monotropic Programming*, Wiley, 1984.

17. P. A. Steenbrink: *Optimization of Transport Networks*, Wiley, London, 1974.

18. L. S. Thakur: "Error analysis for convex separable programs", *SIAM Journal of Applied Mathematics*, 704-714, 1978.