SIAM J. CONTROL AND OPTIMIZATION
Vol. 20, No. 2, March 1982

0363-0129/82/2002-0006 $01.00/0

PROJECTED NEWTON METHODS FOR OPTIMIZATION PROBLEMS WITH SIMPLE CONSTRAINTS*

DIMITRI P. BERTSEKAS†

Abstract. We consider the problem $\min\{f(x) | x \geqq 0\}$, and propose algorithms of the form $x_{k+1} = [x_k - \alpha_k D_k \nabla f(x_k)]^+$, where $[\cdot]^+$ denotes projection on the positive orthant, α_k is a stepsize chosen by an Armijo-like rule and D_k is a positive definite symmetric matrix which is partly diagonal. We show that D_k can be calculated simply on the basis of second derivatives of f so that the resulting Newton-like algorithm has a typically superlinear rate of convergence. With other choices of D_k convergence at a typically linear rate is obtained. The algorithms are almost as simple as their unconstrained counterparts. They are well suited for problems of large dimension such as those arising in optimal control while being competitive with existing methods for low-dimensional problems. The effectiveness of the Newton-like algorithm is demonstrated via computational examples involving as many as 10,000 variables. Extensions to general linearly constrained problems are also provided. These extensions utilize a notion of an active generalized rectangle patterned after the notion of an active manifold used in manifold suboptimization methods. By contrast with these methods, many constraints can be added or subtracted from the binding set at each iteration without the need to solve a quadratic programming problem.

1. Introduction.

We consider the problem

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \geqq 0, \end{aligned} \tag{1}$$

where $f: R^n \to R$ is a continuously differentiable function, and the vector inequality $x \geqq 0$ is meant to be componentwise (i.e., for $x = (x^1, x^2, \cdots, x^n) \in R^n$, we write $x \geqq 0$ if $x^i \geqq 0$ for all $i = 1, \cdots, n$). This type of problem arises very often in applications; for example, when f is a dual functional relative to an original inequality constrained primal problem and x represents a vector of nonnegative Lagrange multipliers corresponding to the inequality constraints, and when f represents an augmented Lagrangian or exact penalty function taking into account other possibly nonlinear equality and inequality constraints. The analysis and algorithms that follow apply also with minor modifications to problems with rectangle constraints such as

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } b_1 \leqq x \leqq b_2, \end{aligned} \tag{2}$$

where b_1 and b_2 are given vectors. Problems (1) and (2) are referred to as *simply constrained problems*, and their algorithmic solution is the primary subject of this paper.

In view of the simplicity of the constraints, one would expect that solution of problem (1) is almost as easy as unconstrained minimization of f. This expectation is partly justified in that the first order necessary condition for a vector $\bar{x} = (\bar{x}^1, \cdots, \bar{x}^n)$ to be a local minimum of problem (1) takes the simple form

$$\frac{\partial f(\bar{x})}{\partial x^i} \geqq 0 \quad \forall i = 1, \cdots, n, \tag{3a}$$

* Received by the editors August 26, 1980, and in final revised form April 22, 1981.

† Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139. This work was supported in part by the National Science Foundation under grant ENG-79-06332 and in part by Alphatech Inc., through Department of Energy contract DE-AC01-79ET29243.

(3b) $$\frac{\partial f(\bar{x})}{\partial x^i} = 0 \quad \text{if } x^i > 0, \quad \forall i = 1, \cdots, n.$$

Furthermore the direct analog of the method of steepest descent takes the simple form

(4) $$x_{k+1} = [x_k - \alpha_k \nabla f(x_k)]^+, \qquad k = 0, 1, \cdots,$$

where α_k is a positive scalar stepsize and for any vector $z = (z^1, \cdots, z^n) \in R^n$ we denote

$$[z]^+ = \begin{bmatrix} \max\{0, z^1\} \\ \vdots \\ \max\{0, z^n\} \end{bmatrix}.$$

The stepwise α_k may be chosen in a number of ways. In the original proposal of Goldstein [1] and Levitin and Poljak [2], α_k is taken to be a constant $\bar{\alpha}$ (i.e., $\alpha_k \equiv \bar{\alpha}$, for all k), and a convergence result is shown under the assumption that $\bar{\alpha}$ is sufficiently small and ∇f is Lipschitz continuous. In general a proper value for $\bar{\alpha}$ can be found only through experimentation. An alternative suggested by McCormick [3] is to choose α_k by function minimization along the arc of points $x_k(\alpha)$, $\alpha \geqq 0$, where

(5) $$x_k(\alpha) = [x_k - \alpha \nabla f(x_k)]^+, \qquad \alpha \geqq 0.$$

Thus α_k is chosen so that

(6) $$f[x_k(\alpha_k)] = \min_{\alpha \geqq 0} f[x_k(\alpha)].$$

Unfortunately the minimization above is very difficult to carry out, particularly for problems of large dimension, since $f[x_k(\alpha)]$ need not be differentiable, convex, or unimodal as a function of α even if f is convex. For most problems we prefer the Armijo-like stepsize rule, first proposed in Bertsekas [4], whereby α_k is given by

(7a) $$\alpha_k = \beta^{m_k} s,$$

where m_k is the first nonnegative integer m satisfying

(7b) $$f(x_k) - f[x_k(\beta^m s)] \geqq \sigma \nabla f(x_k)'[x_k - x_k(\beta^m s)].$$

Here the scalars s, β and σ are fixed and are chosen so that $s > 0$, $\beta \in (0, 1)$ and $\sigma \in (0, \frac{1}{2})$. In addition to being easily implementable and convergent, the algorithm (4), (7) has the advantage that when it converges to a local minimum x^* satisfying the standard second order sufficiency conditions for optimality (including strict complementarity) it identifies the binding constraints at x^* in a finite number of iterations in the sense that there exists $\bar{k}$ such that

(8) $$B(x^*) = B(x_k) \quad \forall k > \bar{k},$$

where, for every $x \in R^m$, $B(x)$ denotes the set of indices of binding constraints at x,

(9) $$B(x) = \{i \mid x^i = 0, i = 1, \cdots, n\}.$$

Minor modifications of the proofs given in [4] show that the results stated above hold also for the algorithm

(10) $$x_{k+1} = [x_k - \alpha_k D_k \nabla f(x_k)]^+,$$

where D_k is a *diagonal* positive definite matrix, and α_k is chosen by (7) where now $x_k(\alpha)$ is given by

(11) $$x_k(\alpha) = [x_k - \alpha D_k \nabla f(x_k)]^+.$$

For this it is necessary to assume that the diagonal elements d_k^i, $i = 1, \cdots, n$ of the matrices D_k satisfy

$$\underline{d} \leqq d_k^i \leqq \bar{d} \quad \forall i = 1, \cdots, n, \quad k = 0, 1, \cdots,$$

where $\underline{d}$ and $\bar{d}$ are some positive scalars.

While it is often possible to achieve substantial computational savings by proper diagonal scaling of ∇f as in (10), the resulting algorithm is typically characterized by linear convergence rate [4], [22]. Any attempt to construct a superlinearly convergent algorithm must by necessity involve a nondiagonal scaling matrix D_k which is an adequate approximation of the inverse Hessian $\nabla^2 f(x_k)^{-1}$, at least along a suitable subspace. At this point we find that the algorithms available at present are far more complicated than their unconstrained counterparts, particularly when the problem has large dimension. Thus the most straightforward extension of Newton's method is given by

$$x_{k+1} = x_k + \alpha_k(\bar{x}_k - x_k), \tag{12}$$

where $\bar{x}_k$ is a solution of the quadratic program

$$\begin{aligned} &\text{minimize } \nabla f(x_k)'(x - x_k) + \tfrac{1}{2}(x - x_k)'\nabla^2 f(x_k)(x - x_k) \\ &\text{subject to } x \geqq 0, \end{aligned} \tag{13}$$

and α_k is a stepsize parameter. There are convergence and superlinear rate of convergence results in the literature regarding this type of method (Levitin and Poljak [2], Dunn [5]) and its quasi-Newton versions (Garcia-Palomares and Mangasarian [6]); however, its effectiveness is strongly dependent upon the computational requirements of solving the quadratic program (13). For problems of small dimension problem (13) can be solved rather quickly by standard pivoting or manifold suboptimization methods, but for large-dimensional problems the solution of the quadratic program (13) by standard methods can be very time consuming. Indeed there are large-scale quadratic programming problems arising in optimal control, the solution of which by pivoting methods is unthinkable. In any case the facility or lack thereof of solving the quadratic program (13) must be accounted for when comparing method (12) against other alternatives.

Another possible approach for constructing superlinearly convergent algorithms for solving problem (1) stems from the original gradient projection proposal of Rosen [7] and is based on manifold suboptimization and active set strategies as in Gill and Murray [8], Goldfarb [9], Luenberger [10] and other sources, (see Lenard [11] for up-to-date performance evaluation of various alternatives). Methods of this type are quite efficient for problems of relatively small dimension, but are typically unattractive for large-scale problems with a large number of constraints binding at a solution. The main reason is that typically at most one constraint can be added to the active set at each iteration, so if, for example, 1,000 constraints are binding at the point of convergence and an interior starting point is selected, then the method will require at least 1,000 iterations (and possibly many more) to converge. While several authors [8], [10] have alluded to the possibility of bending the direction of search along the constraint boundary, the only specific proposal known to the author that has been made in the context of the manifold suboptimization approach is the one of McCormick [12] and it does not seem particularly attractive for large-scale problems. (The quasi-Newton methods proposed by Brayton and Cullum [13] incorporate bending but simultaneously require the solution of quadratic programming subproblems.)

Manifold suboptimization methods require also additional computation overhead in deciding which constraint to drop from the currently active set. For the apparently most successful strategies (Lenard [11]) which attempt to drop as many constraints as possible this overhead can be significant and must be taken into account when comparing the manifold suboptimization approach with other alternatives.

The algorithms proposed in this paper attempt to combine the basic simplicity of the steepest descent iteration (4), (7) with the sophistication and fast convergence of the constrained Newton's method (12), (13). They do not involve solution of a quadratic program thereby avoiding the associated computational overhead, and there is no bound to the number of constraints that can be added to the currently active set thereby bypassing a serious inherent limitation of manifold suboptimization methods. The basic form of the method is

$$x_{k+1} = x_k(\alpha_k), \tag{14}$$

where

$$x_k(\alpha) = [x_k - \alpha D_k \nabla f(x_k)]^+ \quad \forall \alpha \geqq 0. \tag{15}$$

D_k is a positive definite symmetric matrix which is partly diagonal, and α_k is a stepsize determined by an Armijo-like rule similar to (1) that will be described later. The convergence and rate of convergence properties of this method are discussed in § 2. A key property of the method is that under mild assumptions *it identifies the manifold of binding constraints at a solution in a finite number of iterations* in the sence of (8). This means that eventually the method is reduced to an unconstrained method on this manifold and brings to bear the extensive methodology and analysis relating to unconstrained minimization algorithms.

In § 3 we discuss how the method (14), (15) can form the basis for constructing algorithms for general linearly constrained problems of the form

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } b_1 \leqq Ax \leqq b_2. \end{aligned} \tag{16}$$

The main idea here is to view problem (16) *locally* as a simply constrained problem via a transformation of variables. For example, if the matrix A is square and invertible problem (16) is equivalent to the problem

$$\begin{aligned} &\text{minimize } h(y) \triangleq f(A^{-1}y) \\ &\text{subject to } b_1 \leqq y \leqq b_2, \end{aligned}$$

via the transformation

$$y = Ax.$$

A similar approach based on an active set strategy is employed when A is not square and invertible. The ideas are similar to those involved in manifold suboptimization methods where a linear manifold is selected as a "local universe" for the purposes of the current iteration. In our algorithms we take a suitably chosen rectangle (i.e., a set described by upper and lower bounds on the variables) as a "local universe" instead of a manifold.

Finally in § 4 we provide results of computational experiments with large scale optimal control problems some of which involve several thousand variables.

Throughout the paper we emphasize Newton-like methods as prototypes for broad classes of superlinearly converging algorithms that fit the framework of the

paper. We often make positive definiteness assumptions on the Hessian matrix of f in order to avoid getting bogged down in technical details relating to modifications of Newton's method such as those employed in unconstrained minimization [14]–[16] to account for the possibility that $\nabla^2 f$ is not positive definite. Quasi-Newton, approximate Newton and conjugate gradient versions of the Newton-like methods presented are possible but the discussion of specific implementations is beyond the scope of the paper. More generally it may be said that the nature of the algorithms proposed is such that almost every useful idea from unconstrained minimization can be fruitfully adapted within the constrained minimization framework considered here; however, the precise details of how this should be done may involve considerable further research and experimentation.

The notation employed throughout the paper is as follows. All vectors are considered to be column vectors. A prime denotes transposition. The standard norm in R^n is denoted by $|\cdot|$, i.e., for $x=(x^1,\cdots,x^n)$ we write $|x|=[\sum_{i=1}^n (x^i)^2]^{1/2}$. The gradient and Hessian of a function $f: R^n \to R$ are denoted by ∇f and $\nabla^2 f$ respectively.

2. Algorithms for minimization subject to simple constraints. We consider first the problem $\min\{f(x) | x \geqq 0\}$ of (1). Any vector $\bar{x} \geqq 0$ satisfying the first order necessary condition (3) will be referred to as a *critical point with respect to problem* (1). We focus attention at iterations of the form

$$x_{k+1} = [x_k - \alpha_k D_k \nabla f(x_k)]^+,$$

where D_k is a positive definite symmetric matrix and α_k is chosen by search along the arc of points

$$x_k(\alpha) = [x_k - \alpha D_k \nabla f(x_k)]^+, \qquad \alpha \geqq 0.$$

It is easy to construct examples (see Fig. 1) where an arbitrary positive definite choice of the matrix D_k leads to situations where it is impossible to reduce the value of the objective by suitable choice of the stepsize α (i.e., $f[x_k(\alpha)] \geqq f(x_k)$, $\forall \alpha \geqq 0$). The

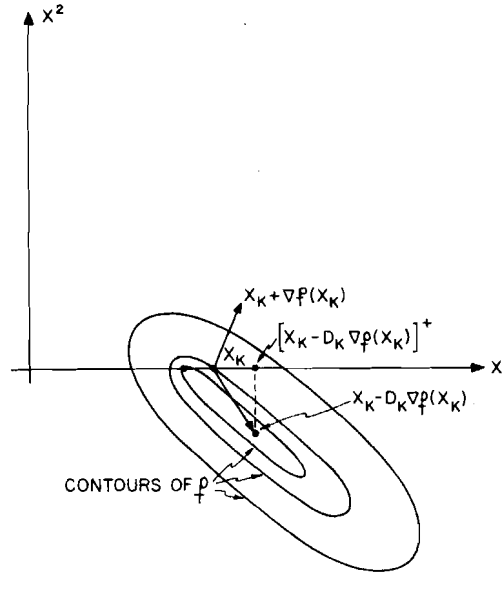

FIG. 1

following proposition identifies a class of matrices D_k for which an objective function reduction is possible. Define for all $x \geqq 0$,

$$I^+(x) = \left\{ i \,\middle|\, x^i = 0, \frac{\partial f(x)}{\partial x^i} > 0 \right\}. \tag{17}$$

We say that a symmetric $n \times n$ matrix D with elements d^{ij} *is diagonal with respect to a subset of indices* $I \subset \{1, 2, \cdots, n\}$ if

$$d^{ij} = 0 \quad \forall i \in I, \quad j = 1, 2, \cdots, n, \quad j \neq i. \tag{18}$$

PROPOSITION 1. *Let* $x \geqq 0$ *and* D *be a positive definite symmetric matrix which is diagonal with respect to* $I^+(x)$ *and denote*

$$x(\alpha) = [x - \alpha D \nabla f(x)]^+ \quad \forall \alpha \geqq 0. \tag{19}$$

(a) *The vector* x *is a critical point with respect to problem* (1) *if and only if*

$$x = x(\alpha) \quad \forall \alpha \geqq 0.$$

(b) *If* x *is not a critical point with respect to problem* (1) *there exists a scalar* $\bar{\alpha} > 0$ *such that*

$$f[x(\alpha)] < f(x) \quad \forall \alpha \in (0, \bar{\alpha}]. \tag{20}$$

Proof. Assume without loss of generality that for some integer r we have

$$I^+(x) = \{r+1, \cdots, n\}.$$

Then D has the form

$$D = \begin{bmatrix} \bar{D} & 0 \\ \hline & d^{r+1} \quad\quad 0 \\ 0 & \ddots \\ & 0 \quad\quad d^n \end{bmatrix}, \tag{21}$$

where $\bar{D}$ is positive definite and $d^i > 0$, $i = r+1, \cdots, n$. Denote

$$p = D \nabla f(x). \tag{22}$$

(a) Assume x is a critical point. Then using (3), (17)

$$\frac{\partial f(x)}{\partial x^i} = 0 \quad \forall i = 1, \cdots, r,$$

$$\frac{\partial f(x)}{\partial x^i} > 0 \quad \text{if } x^i = 0, \quad \forall i = r+1, \cdots, n.$$

These relations and the positivity of d^i, $i = r+1, \cdots, n$ imply that

$$p^i = 0 \quad \forall i = 1, \cdots, r,$$

$$p^i > 0 \quad \forall i = r+1, \cdots, n.$$

Since $x^i(\alpha) = [x^i - \alpha p^i]^+$ and $x^i = 0$ for $i = r+1, \cdots, n$ it follows that $x^i(\alpha) = x^i$ for all i, and $\alpha \geqq 0$.

Conversely assume that $x = x(\alpha)$ for all $\alpha \geqq 0$. Then we must have

$$p^i = 0 \quad \forall i = 1, \cdots, n \quad \text{with } x^i > 0,$$

$$p^i \geqq 0 \quad \forall i = 1, \cdots, n \quad \text{with } x^i = 0.$$

Now by definition of $I^+(x)$ we have that if $x^i = 0$ and $i \notin I^+(x)$ then $\partial f(x)/\partial x^i \leqq 0$. This together with the relations above imply

$$\sum_{i=1}^{r} p^i \frac{\partial f(x)}{\partial x^i} \leqq 0.$$

Since by (21), (22),

$$\begin{bmatrix} p_1 \\ \vdots \\ p_r \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f(x)}{\partial x^1} \\ \vdots \\ \dfrac{\partial f(x)}{\partial x^r} \end{bmatrix}$$

and $\bar{D}$ is positive definite we have $\sum_{i=1}^{r} p^i \partial f(x)/\partial x^i \geqq 0$, and it follows that

$$p^i = \frac{\partial f(x)}{\partial x^i} = 0 \quad \forall i = 1, \cdots, r.$$

Since for $i = r+1, \cdots, n$, $\partial f(x)/\partial x^i > 0$ and $x^i = 0$, we obtain that x is a critical point.

(b) For $i = r+1, \cdots, n$ we have $\partial f(x)/\partial x^i > 0$, $x^i = 0$, and, from (21), (22), $p^i > 0$. Since $x^i(\alpha) = [x^i - \alpha p^i]^+$ we obtain

$$x^i = x^i(\alpha) = 0 \quad \forall \alpha \geqq 0, \quad i = r+1, \cdots, n. \tag{23}$$

Consider the sets of indices

$$I_1 = \{i \mid x^i > 0 \text{ or } x^i = 0 \text{ and } p^i < 0, i = 1, \cdots, r\}, \tag{24}$$

$$I_2 = \{i \mid x^i = 0 \text{ and } p^i \geqq 0, i = 1, \cdots, r\}. \tag{25}$$

Let

$$\alpha_1 = \sup \{\alpha \mid x^i - \alpha p^i \geqq 0, \forall i \in I_1\}. \tag{26}$$

Note that, in view of the definition of I_1, α_1 is either positive or $+\infty$. Define the vector $\bar{p}$ with coordinates

$$\bar{p}^i = \begin{cases} p^i & \text{if } i \in I_2, \\ 0 & \text{if } i \in I_2 \text{ or } i \in I^+(x) \end{cases} \tag{27}$$

In view of (23)–(27), we have

$$x(\alpha) = x - \alpha \bar{p} \quad \forall \alpha \in (0, \alpha_1). \tag{28}$$

In view of (25) and the definition of $I^+(x)$, we have

$$\frac{\partial f(x)}{\partial x^i} \leqq 0 \quad \forall i \in I_2, \tag{29}$$

and hence

$$\sum_{i \in I_2} \frac{\partial f(x)}{\partial x^i} p^i \leqq 0. \tag{30}$$

Now using (27) and (30), we have

$$\nabla f(x)' \bar{p} = \sum_{i \in I_1} \frac{\partial f(x)}{\partial x^i} p^i \geqq \sum_{i=1}^{r} \frac{\partial f(x)}{\partial x^i} p^i. \tag{31}$$

Since x is not a critical point, by part (a) and (28), we must have $x \neq x(\alpha)$ for some HH$\alpha > 0$ and hence also, in view of (23), $p^i \neq 0$ for some $i \in \{1, \cdots, r\}$. In view of the positive definiteness of $\bar{D}$ and (21), (22) it follows that

$$\sum_{i=1}^{r} \frac{\partial f(x)}{\partial x^i} p^i > 0.$$

It follows from (31) that

$$\nabla f(x)' \bar{p} > 0.$$

Combining this relation with (28) and the fact $\alpha_1 > 0$ yields that $\bar{p}$ is a feasible descent direction at x and there exists a scalar $\bar{\alpha} > 0$ for which the desired relation (20) is satisfied. Q.E.D.

Based on Proposition 1 we are led to the conclusion that the matrix D_k in the iteration

$$x_{k+1} = [x_k - \alpha_k D_k \nabla f(x_k)]^+$$

should be chosen diagonal with respect to a subset of indices that contains

$$I^+(x_k) = \left\{ i \,\middle|\, x_k^i = 0, \frac{\partial f(x_k)}{\partial x^i} > 0 \right\}.$$

Unfortunately the set $I^+(x_k)$ exhibits an undesirable discontinuity at the boundary of the constraint set, whereby given a sequence $\{x_k\}$ of interior points that converges to a boundary point $\bar{x}$ the set $I^+(x_k)$ may be strictly smaller than the set $I^+(\bar{x})$. This causes difficulties in proving convergence of the algorithm and may have an adverse effect on its rate of convergence. (This phenomenon is quite common in feasible direction algorithms and is referred to as zigzagging or jamming.) For this reason we will employ certain enlargements of the sets $I^+(x_k)$ with the aim of bypassing these difficulties.

The algorithm that we describe utilizes a scalar $\varepsilon > 0$ (typically small), a fixed[1] *diagonal* positive definite matrix M (for example the identity), and two parameters $\beta \in (0, 1)$ and $\sigma \in (0, \frac{1}{2})$ that will be used in connection with an Armijo-like stepsize rule. An initial vector $x_0 \geqq 0$ is chosen and at the kth iteration of the algorithm we have a vector $x_k \geqq 0$. Denote

$$w_k = |x_k - [x_k - M\nabla f(x_k)]^+|, \qquad \varepsilon_k = \min\{\varepsilon, w_k\}.$$

$(k+1)$st iteration of the Algorithm. We select a positive definite symmetric matrix D_k which is diagonal with respect to the set I_k^+ given by

$$I_k^+ = \left\{ i \,\middle|\, 0 \leqq x_k^i \leqq \varepsilon_k, \frac{\partial f(x_k)}{\partial x^i} > 0 \right\}. \tag{32}$$

Denote

$$p_k = D_k \nabla f(x_k), \tag{33}$$

$$x_k(\alpha) = [x_k - \alpha p_k]^+ \quad \forall \alpha \geqq 0. \tag{34}$$

Then x_{k+1} is given by

$$x_{k+1} = x_k(\alpha_k), \tag{35}$$

[1] Actually the results that follow can be shown also for the case where M is changed from one iteration to the next in a way that its diagonal elements are bounded above and away from zero.

where

$$\alpha_k = \beta^{m_k} \tag{36}$$

and m_k is the first nonnegative integer m such that[2]

$$f(x_k) - f[x_k(\beta^m)] \geq \sigma\left\{\beta^m \sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i + \sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i}[x_k^i - x_k^i(\beta^m)]\right\}. \tag{37}$$

The stepsize rule (36), (37) (see Fig. 2) may be viewed as a combination of the Armijo-like rule (7) and the Armijo rule usually employed in unconstrained minimization (see, e.g., Polak [18]). When I_k^+ is empty the right-hand side of (37) becomes $\sigma\beta^m \nabla f(x_k)' p_k$ and is identical to the corresponding expression of the Armijo

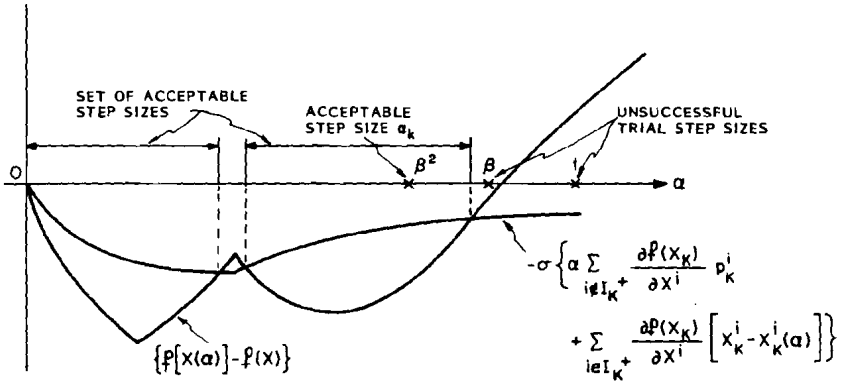

FIG. 2

rule in unconstrained optimization, while if $I_k^+ = \{1, 2, \cdots, n\}$ then inequality (37) is identical with (7). Note that for all k we have

$$I_k^+ \supset I^+(x_k)$$

so the matrix D_k is diagonal with respect to $I^+(x_k)$. It is possible to show that for all $m \geq 0$ the right-hand side of (37) is nonnegative, and it is positive if and only if x_k is not a critical point. Indeed since D_k is positive definite and diagonal with respect to I_k^+ we have

$$\sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i \geq 0 \quad \forall k = 0, 1, \cdots, \tag{38}$$

while for all $i \in I_k^+$, in view of the fact $\partial f(x^k)/\partial x^i > 0$, we have $p_k^i > 0$ and hence

$$x_k^i - x_k^i(\alpha) \geq 0 \quad \forall \alpha \geq 0, \quad i \in I_k^+, \quad k = 0, 1, \cdots,$$

$$\frac{\partial f(x_k)}{\partial x^i}[x_k^i - x_k^i(\alpha)] \geq 0 \quad \forall \alpha \geq 0, \quad i \in I_k^+, \quad k = 0, 1, \cdots. \tag{39}$$

This shows that the right side of (37) is nonnegative. If x_k is not critical then it is easily seen (compare also with the proof of Proposition 1(b)) that one of the inequalities

[2] The results that follow can also be proved if $\sum_{i \in I_k^+} (\partial f(x_k)/\partial x^i) p_k^i$ is replaced in (37) by $\gamma_k \sum_{i \in I_k} (\partial f(x_k)/\partial x^i) p_k^i$, where $\gamma_k = \min\{1, \bar{\alpha}_k\}$ and $\bar{\alpha}_k = \sup\{\alpha | x_k^i - \alpha p_k^i \geq 0, \forall i \notin I_k^+\}$. This modification makes (37) easier to satisfy.

(38) or (39) is strict for $\alpha > 0$ so the right side of (37) is positive for all $m \geqq 0$. A slight modification of the proof of Proposition 1(b) also shows that if x_k is not a critical point then (37) will be satisfied for all m sufficiently large so the stepsize α_k is well defined and can be determined via a finite number of arithmetic operations. If x_k is a critical point then, by Proposition 1(a), we have $x_k = x_k(\alpha)$ for all $x \geqq 0$. Furthermore the argument given in the proof of Proposition 1(a) shows that

$$\sum_{i=1}^{r} \frac{\partial f(x_k)}{\partial x^i} p_k^i = 0$$

so both terms in the right side of (37) are zero. Since also $x_k = x_k(\alpha)$ for all $\alpha \geqq 0$ it follows that (37) is satisfied for $m = 0$ thereby implying that

$$x_{k+1} = x_k(1) = x_k \quad \text{if } x_k \text{ is critical.}$$

In conclusion, the algorithm is well defined, decreases the value of the objective function at each iteration k for which x_k is not a critical point, and essentially terminates if x_k is critical. We proceed to analyze its convergence and rate of convergence properties. To this end we will make use of the following two assumptions:

(A) *The gradient ∇f is Lipschitz continuous on each bounded set of R^n; i.e., given any bounded set $S \subset R^n$ there exists a scalar L (depending on S) such that*

$$|\nabla f(x) - \nabla f(y)| \leqq L|x - y| \quad \forall x, y \in S. \tag{40}$$

(B) *There exist positive scalars λ_1, λ_2 and nonnegative integers q_1, q_2 such that*

$$\lambda_1 w_k^{q_1} |z|^2 \leqq z' D_k z \leqq \lambda_2 w_k^{q_2} |z|^2 \quad \forall z \in R^n, \quad k = 0, 1 \cdots, \tag{41}$$

where

$$w_k = |x_k - [x_k - M \nabla f(x_k)]^+|.$$

Assumption (A) is not essential for the result of Proposition 2 that follows, but simplifies its proof. It is satisfied for just about every problem likely to appear in practice. For example it is satisfied when f is twice differentiable as well as when f is an augmented Lagrangian of the type used for inequality constrained problems involving twice differentiable functions. Assumption (B) is a condition of the type commonly utilized in connection with unconstrained minimization algorithms. When $q_1 = q_2 = 0$, relation (41) takes the form

$$\lambda_1 |z|^2 \leqq z' D_k z \leqq \lambda_2 |z|^2 \quad \forall \in R^n, \quad k = 0, 1 \cdots \tag{42}$$

and simply says that the eigenvalues of D_k are uniformly bounded above and away from zero.

PROPOSITION 2. *Under Assumptions* (A) *and* (B) *above, every limit point of a sequence $\{x_k\}$ generated by iteration* (35) *is a critical point with respect to problem* (1).

Proof. Assume the contrary, i.e., that there exists a subsequence $\{x_k\}_K$ converging to a vector $\bar{x}$ which is not critical. Since $\{f(x_k)\}$ is decreasing and f is continuous it follows that $\{f(x_k)\}$ converges to $f(\bar{x})$ and therefore

$$[f(x_k) - f(x_{k+1})] \to 0.$$

Since each of the sums in the right-hand side of (37) is nonnegative (cf. (38), (39)), we must have

$$\alpha_k \sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i \to 0, \tag{43}$$

$$\sum_{i\in I_k^+} \frac{\partial f(x_k)}{\partial x^i}[x_k^i - x_k^i(\alpha_k)] \to 0. \tag{44}$$

Also since $\bar{x}$ is not critical and M is diagonal we have $|\bar{x} - [\bar{x} - M\nabla f(\bar{x})]^+| \neq 0$, so (41) implies that the eigenvalues of $\{D_k\}_K$ are uniformly bounded above and away from zero. In view of the fact that D_k is diagonal with respect to I_k^+, it follows that there exist positive scalars $\bar{\lambda}_1, \bar{\lambda}_2$ such that for all $k \in K$ that are sufficiently large

$$0 < \bar{\lambda}_1 \frac{\partial f(x_k)}{\partial x^i} \leqq p_k^i \leqq \bar{\lambda}_2 \frac{\partial f(x_k)}{\partial x^i} \quad \forall i \in I_k^+, \tag{45}$$

$$\bar{\lambda}_1 \sum_{i\in I_k^+} \left|\frac{\partial f(x_k)}{\partial x^i}\right|^2 \leqq \sum_{i\in I_k^+} p_k^i \frac{\partial f(x_k)}{\partial x^i} \leqq \bar{\lambda}_2 \sum_{i\in I_k^+} \left|\frac{\partial f(x_k)}{\partial x^i}\right|^2. \tag{46}$$

We will show that our hypotheses so far lead to the conclusion that

$$\lim_{\substack{k\to\infty \\ k\in K}} \inf \alpha_k = 0. \tag{47}$$

Indeed, since $\bar{x}$ is not a critical point there must exist an index i such that either

$$\bar{x}^i > 0 \quad \text{and} \quad \frac{\partial f(\bar{x})}{\partial x^i} \neq 0 \tag{48}$$

or

$$\bar{x}^i = 0 \quad \text{and} \quad \frac{\partial f(\bar{x})}{\partial x^i} < 0. \tag{49}$$

If $i \notin I_k^+$ for an infinite number of indices $k \in K$ then (47) follows from (43), (46), (48) and (49). If $i \in I_k^+$ for an infinite number of indices $k \in K$ then for all those indices we must have $\partial f(x_k)/\partial x^i > 0$ so (49) cannot hold. Therefore from (48)

$$\bar{x}^i > 0 \quad \text{and} \quad \frac{\partial f(\bar{x})}{\partial x^i} > 0. \tag{50}$$

Since we have [cf. (39)] for all $k \in K$ for which $i \in I_k^+$

$$\sum_{j\in I_k^+} \frac{\partial f(x_k)}{\partial x^j}[x_k^j - x_k^j(\alpha_k)] \geqq \frac{\partial f(x_k)}{\partial x^i}[x_k^i - x_k^i(\alpha_k)] \geqq 0,$$

it follows from (44) and (50) that

$$\lim_{\substack{k\to\infty \\ k\in K}} [x_k^i - x_k^i(\alpha_k)] = 0.$$

Using the above relation, (45) and (50), we obtain (47).

We will complete the proof by showing that $\{\alpha_k\}_K$ is bounded away from zero thereby contradicting (47). Indeed in view of (46) the subsequences $\{x_k\}_K$, $\{p_k\}_K$ and $\{x_k(\alpha)\}_K$, $\alpha \in [0, 1]$ are uniformly bounded so by Assumption (A) there exists a scalar $L > 0$ such that for all $t \in [0, 1]$, $\alpha \in [0, 1]$ and $k \in K$ we have

$$|\nabla f(x_k) - \nabla f[x_k - t[x_k - x_k(\alpha)]]| \leqq tL|x_k - x_k(\alpha)|. \tag{51}$$

We have for all $k \in K$ and $\alpha \in [0, 1]$

$$f[x_k(\alpha)] = f(x_k) + \nabla f(x_k)'[x_k(\alpha) - x_k] + \int_0^1 \{\nabla f(x_k) - \nabla f[x_k - t[x_k - x_k(\alpha)]]\}' \, dt \, [x_k - x_k(\alpha)],$$

so

$$f(x_k)-f[x_k(\alpha)]$$

$$=\nabla f(x_k)'[x_k-x_k(\alpha)]+\int_0^1 \{\nabla f[x_k-t[x_k-x_k(\alpha)]]-\nabla f(x_k)\}'\,dt[x_k-x_k(\alpha)]$$

$$\geqq \nabla f(x_k)'[x_k-x_k(\alpha)]-\int_0^1 |\nabla f[x_k-t[x_k-x_k(\alpha)]]-\nabla f(x_k)|\,dt|x_k-x_k(\alpha)|,$$

and finally by using (51)

$$f(x_k)-f[x_k(\alpha)]\geqq \nabla f(x_k)'[x_k-x_k(\alpha)]-\frac{L}{2}|x_k-x_k(\alpha)|^2. \tag{52}$$

For $i\in I_k^+$ we have $x_k^i(\alpha)=[x_k^i-\alpha p_k^i]^+\geqq x_k^i-\alpha p_k^i$ and $p_k^i>0$, so $0\leqq x_k^i-x_k^i(\alpha)\leqq \alpha p_k^i$. It follows using (45) that

$$\sum_{i\in I_k^+}|x_k^i-x_k^i(\alpha)|^2\leqq \alpha\sum_{i\in I_k^+}p_k^i[x_k^i-x_k^i(\alpha)]\leqq \alpha\bar{\lambda}_2\sum_{i\in I_k^+}\frac{\partial f(x_k)}{\partial x^i}[x_k^i-x_k^i(\alpha)]. \tag{53}$$

Consider the sets

$$I_{1,k}=\left\{i\,\middle|\,\frac{\partial f(x_k)}{\partial x^i}>0,\ i\notin I_k^+\right\},\qquad I_{2,k}=\left\{i\,\middle|\,\frac{\partial f(x_k)}{\partial x^i}\leqq 0,\ i\notin I_k^+\right\}.$$

For all $i\in I_{1,k}$ we must have $x_k^i>\varepsilon_k$ for otherwise we would have $i\in I_k^+$. Since $|\bar{x}-[\bar{x}-M\nabla f(\bar{x})]^+|\neq 0$ we must have $\lim_{k\to 0,k\in K}\inf\varepsilon_k>0$ and $\varepsilon_k>0$ for all k. Let $\bar{\varepsilon}>0$ be such that $\bar{\varepsilon}\leqq\varepsilon_k$ for all $k\in K$, and let B be such that $|p_k^i|\leqq B$ for all i and $k\in K$. Then for all $\alpha\in[0,\bar{\varepsilon}/B]$ we have $x_k^i(\alpha)=x_k^i-\alpha p_k^i$ for all $i\in I_{i,k}$ so it follows that

$$\sum_{i\in I_{1,k}}\frac{\partial f(x_k)}{\partial x^i}[x_k^i-x_k^i(\alpha)]=\alpha\sum_{i\in I_{1,k}}\frac{\partial f(x_k)}{\partial x^i}p_k^i\quad\forall\alpha\in\left[0,\frac{\bar{\varepsilon}}{B}\right]. \tag{54}$$

Also for all $\alpha\geqq 0$ we have $x_k^i-x_k^i(\alpha)\leqq\alpha p_k^i$, and since $\partial f(x_k)/\partial x^i\leqq 0$ for all $i\in I_{2,k}$, we obtain

$$\sum_{i\in I_{2,k}}\frac{\partial f(x_k)}{\partial x^i}[x_k^i-x_k^i(\alpha)]\geqq\alpha\sum_{i\in I_{2,k}}\frac{\partial f(x_k)}{\partial x^i}p_k^i. \tag{55}$$

Combining (54) and (55), we obtain

$$\sum_{i\notin I_k^+}\frac{\partial f(x_k)}{\partial x^i}[x_k^i-x_k^i(\alpha)]\geqq\alpha\sum_{i\notin I_k^+}\frac{\partial f(x_k)}{\partial x^i}p_k^i\quad\forall\alpha\in\left[0,\frac{\bar{\varepsilon}}{B}\right]. \tag{56}$$

For all $\alpha\geqq 0$ we also have

$$|x_k^i-x_k^i(\alpha)|\leqq\alpha|p_k^i|\quad\forall i=1,\cdots,n.$$

Furthermore it is easily seen using Assumption (B) that there exists $\lambda>0$ such that

$$\sum_{i\notin I_k^+}(p_k^i)^2\leqq\lambda\sum_{i\notin I_k^+}\frac{\partial f(x_k)}{\partial x^i}p_k^i\quad\forall k\in K.$$

Combining the last two relations we obtain for all $\alpha\geqq 0$

$$\sum_{i\notin I_k^+}|x_k^i-x_k^i(\alpha)|^2\leqq\alpha^2\lambda\sum_{i\notin I_k^+}\frac{\partial f(x_k)}{\partial x^i}p_k^i\quad\forall k\in K. \tag{57}$$

We now combine (52), (53), (56) and (57) to obtain for all $\alpha \in [0, (\bar{\varepsilon}/B)]$ and $k \in K$

$$f(x_k) - f[x_k(\alpha)] \geqq \left(\alpha - \frac{\alpha^2 \lambda L}{2}\right) \sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i + \left(1 - \frac{\alpha \bar{\lambda}_2 L}{2}\right) \sum_{i \in I_k^-} \frac{\partial f(x_k)}{\partial x^i} [x_k^i - x_k^i(\alpha)]. \tag{58}$$

Suppose α is chosen so that

$$0 \leqq \alpha \leqq \frac{\bar{\varepsilon}}{B}, \quad 1 - \frac{\alpha \lambda L}{2} \geqq \sigma, \quad 1 - \frac{\alpha \bar{\lambda}_2 L}{2} \geqq \sigma, \quad \alpha \leqq 1 \tag{59}$$

or equivalently

$$0 \leqq \alpha \leqq \min\left\{\frac{\bar{\varepsilon}}{B}, \frac{2(1-\sigma)}{\lambda L}, \frac{2(1-\sigma)}{\bar{\lambda}_2 L}, 1\right\}. \tag{60}$$

Then we have from (58), (59) for all $k \in K$

$$f(x_k) - f[x_k(\alpha)] \geqq \sigma\left\{\alpha \sum_{i \in I_k^+} \frac{\partial f(x_k)}{\partial x^i} p_k^i + \sum_{i \in I_k^-} \frac{\partial f(x_k)}{\partial x^i} [x_k^i - x_k^i(\alpha)]\right\}.$$

This means that if (60) is satisfied with $\beta^m = \alpha$, then the inequality (37) of the Armijo-like rule will be satisfied. It follows from the way the stepsize is reduced that α_k satisfies

$$\alpha_k \geqq \beta \min\left\{\frac{\bar{\varepsilon}}{B}, \frac{2(1-\sigma)}{\lambda L}, \frac{2(1-\sigma)}{\bar{\lambda}_2 L}, 1\right\} \quad \forall k \in K. \tag{61}$$

This contradicts (47) and proves the proposition. Q.E.D.

It is interesting to note that the argument of the last part of the proof above shows that if the level set $\{x \mid f(x) \leqq f(x_0), x \geqq 0\}$ is bounded, then there exists a scalar $\bar{\alpha} > 0$ such that, for every $\alpha \in (0, \bar{\alpha}]$, the *constant* stepsize algorithm $x_{k+1} = x_k(\alpha)$ generates sequences $\{x_k\}$ the limit points of which are critical points with respect to problem (1).

We now focus attention at a local minimum x^* satisfying the following second order sufficiency conditions. For all $x \geqq 0$ we denote by $B(x)$ the set of indices of binding constraints at x, i.e.,

$$B(x) = \{i \mid x^i = 0\} \quad \forall x \geqq 0. \tag{62}$$

(C) *The local minimum x^* of problem* (1) *is such that for some $\delta > 0$, f is twice continuously differentiable in the open sphere $\{x \| x - x^*| < \delta\}$, and there exist positive scalars m_1, m_2 such that*

$$m_1 |z|^2 \leqq z' \nabla^2 f(x) z \leqq m_2 |z|^2 \quad \forall x \text{ such that } |x - x^*| < \delta \text{ and } z \neq 0 \text{ such that } z^i = 0, \forall i \in B(x^*). \tag{63}$$

Furthermore

$$\frac{\partial f(x^*)}{\partial x^i} > 0 \quad \forall i \in B(x^*). \tag{64}$$

The following proposition demonstrates an important property of the algorithm, namely that under mild conditions *it is attracted by a local minimum x^* satisfying Assumption* (C) *and identifies the set of active constraints at x^* in a finite number of iterations. Thus if the algorithm converges to x^* then after a finite number of iterations it is equivalent to an unconstrained optimization method restricted on the subspace of*

binding constraints at x^*. This property is instrumental in proving superlinear convergence of the algorithm when the portion of D_k corresponding to the indices $i \notin I_k^+$ is chosen in a way that approximates the inverse of the portion of the Hessian of f corresponding to these same indices.

PROPOSITION 3. *Let* x^* *be a local minimum of problem* (1) *satisfying Assumption* (C). *Assume also that* (B) *holds in the stronger form whereby, in addition to* (41), *the diagonal elements* d_{ii}^k *of the matrices* D_k *satisfy for some scalar* $\bar{\lambda}_1 > 0$

$$\bar{\lambda}_1 \leqq d_{ii}^k \quad \forall k = 0, 1, \cdots, \quad i \in I_k^+. \tag{65}$$

Then there exists a scalar $\bar{\delta} > 0$ *such that if* $\{x_k\}$ *is a sequence generated by iteration* (35) *and for some index* $\bar{k}$ *we have*

$$|x_{\bar{k}} - x^*| \leqq \bar{\delta}, \tag{66}$$

then $\{x_k\}$ *converges to* x^* *and we have*

$$I_k^+ = B(x_k) = B(x^*) \quad \forall k \geqq \bar{k} + 1. \tag{67}$$

Proof. Since f is twice differentiable on $\{x \mid |x - x^*| < \delta\}$, it follows that there exist scalars $L > 0$ and $\delta_1 \in (0, \delta]$ such that for all x, y with $|x - x^*| \leqq \delta$, $|y - x^*| \leqq \delta_1$ we have

$$|\nabla f(x) - \nabla f(y)| \leqq L|x - y|.$$

Also for x_k sufficiently close to x^* the scalar

$$w_k = |x_k - [x_k - M \nabla f(x_k)]^+|$$

is arbitrarily close to zero while, in view of (64), we have

$$\left[x_k^i - \mu^i \frac{\partial f(x_k)}{\partial x^i} \right]^+ = 0 \quad \forall i \in B(x^*),$$

where μ^i is the ith diagonal element of M. It follows that for x_k sufficiently close to x^* we have

$$x_k^i \leqq w_k = \varepsilon_k \quad \forall i \in B(x^*),$$

while

$$x_k^i > \varepsilon_k \quad \forall i \notin B(x^*).$$

This implies that there exists $\delta_2 \in (0, \delta_1]$ such that

$$B(x^*) = I_k^+ \quad \forall k \text{ such that } |x_k - x^*| \leqq \delta_2. \tag{68}$$

Also there exist scalars $\bar{\varepsilon} > 0$ and $\delta_3 \in (0, \delta_2]$ such that

$$x_k^i > \bar{\varepsilon} \quad \forall i \notin B(x^*) \text{ and } k \text{ such that } |x_k - x^*| \leqq \delta_3.$$

By essentially repeating the argument in the proof of Proposition 2 that led to (61), we find that there exists a scalar $\bar{\alpha} > 0$ such that

$$\alpha_k \geqq \bar{\alpha} \quad \forall k \text{ such that } |x_k - x^*| \leqq \delta_3. \tag{69}$$

By using (65) and (68) it follows that

$$0 < \bar{\lambda}_1 \frac{\partial f(x_k)}{\partial x^i} \leqq p_k^i \quad \forall i \in B(x^*) \text{ and } k \text{ such that } |x_k - x^*| \leqq \delta_3, \tag{70}$$

while there exists a scalar $\lambda > 0$ such that

$$\sum_{i \notin B(x^*)} |p_k^i|^2 \leqq \lambda \sum_{i \notin B(x^*)} \left| \frac{\partial f(x_k)}{\partial x^i} \right|^2 \quad \forall k \text{ such that } |x_k - x^*| \leqq \delta_3. \tag{71}$$

Since $\partial f(x^*)/\partial x^i > 0$ for all $i \in B(x^*)$ and $\partial f(x^*)/\partial x^i = 0$ for all $i \notin B(x^*)$ it follows from (68)–(71) that there exists a scalar $\delta_4 \in (0, \delta_3]$ such that

$$B(x^*) = B(x_{k+1}) \quad \forall k \text{ such that } |x_k - x^*| \leqq \delta_4 \tag{72}$$

and

$$|x_{k+1} - x^*| \leqq \delta_3 \quad \forall k \text{ such that } |x_k - x^*| \leqq \delta_4. \tag{73}$$

In view of (68) we obtain from (72), (73)

$$B(x^*) = B(x_{k+1}) = I_{k+1}^+ \quad \forall k \text{ such that } |x_k - x^*| \leqq \delta_4. \tag{74}$$

Thus when $|x_k - x^*| \leqq \delta_4$ we have $|x_{k+1} - x^*| \leqq \delta_3$, $B(x^*) = B(x_{k+1})$, and the $(k+1)$st iteration of the algorithm reduces to an iteration of an unconstrained minimization algorithm on the subspace of binding constraints at x^*. From known results on unconstrained minimization algorithms (cf. [19, Proposition 1.12]) and Assumption (C) it follows that there exists an (open) neighborhood $N(x^*)$ of x^* such that $|x - x^*| < \delta_4$ for all $x \in N(x^*)$ and with the property that if $x_{k+1} \in N(x^*)$ and $B(x_{k+1}) = B(x^*)$, then $x_{k+2} \in N(x^*)$ and, by (74), $B(x_{k+2}) = B(x^*)$. This argument can be repeated and shows that if for some $\bar{k}$ we have

$$x_{\bar{k}} \in N(x^*), \qquad B(x_{\bar{k}}) = B(x^*),$$

then $\{x_k\} \to x^*$ and

$$x_k \in N(x^*), \qquad B(x_k) = B(x^*) \quad \forall k \geqq \bar{k}.$$

To complete the proof it is sufficient to show that there exists $\bar{\delta} > 0$ such that if $|x_k - x^*| \leqq \bar{\delta}$ then $x_{k+1} \in N(x^*)$ and $B(x_{k+1}) = B(x^*)$. Indeed by repeating the argument that led to (73) and (74), we find that given any $\tilde{\delta} > 0$ there exists a $\bar{\delta} > 0$ such that if $|x_k - x^*| \leqq \bar{\delta}$ then

$$|x_{k+1} - x^*| \leqq \tilde{\delta}, \qquad I_{k+1}^+ = B(x_{k+1}) = B(x^*).$$

By taking $\tilde{\delta}$ sufficiently small so that

$$\{x \mid |x - x^*| \leqq \tilde{\delta}\} \subset N(x^*)$$

the proof is complete. Q.E.D

Under the assumptions of Proposition 3 we see that if the algorithm converges to a local minimum x^* satisfying Assumption (C) then it reduces eventually to an unconstrained minimization method restricted to the subspace

$$T = \{x \mid x^i = 0, \forall i \in B(x^*)\} \tag{75}$$

Furthermore, as shown in Proposition 3, for some index $\bar{k}$ we will have

$$I_k^+ = B(x^*) \quad \forall k \geqq \bar{k}. \tag{76}$$

This shows that if the portion of the matrix D_k corresponding to the indices $i \notin I_k^+$ is chosen to be the inverse of the Hessian of f with respect to these indices then the algorithm eventually reduces to Newton's method restricted on the subspace T.

More specifically, by rearranging indices if necessary, assume without loss of generality that

$$I_k^+ = \{r_k + 1, \cdots, n\}, \tag{77}$$

where r_k is some integer. Then D_k has the form

$$D_k = \begin{bmatrix} \bar{D}_k & \vdots & 0 \\ \cdots & \cdots & \cdots \\ & d_k^{r_k+1} & \\ 0 & \vdots & \ddots \\ & 0 & d_k^n \end{bmatrix}, \tag{78}$$

where $d_k^i > 0$, $i = r_k + 1, \cdots, n$ and $\bar{D}_k$ can be an arbitrary positive definite matrix. Suppose we choose $\bar{D}_k$ to be the inverse of the Hessian of f with respect to the indices $i = 1, \cdots, r_k$, i.e., the elements $[\bar{D}_k^{-1}]_{ij}$ are

$$[\bar{D}_k^{-1}]_{ij} = \frac{\partial^2 f(x_k)}{\partial x^i \partial x^j} \quad \forall\, i, j \notin I_k^+. \tag{79}$$

By Assumption (C), $\nabla^2 f(x^*)$ is positive definite on T so it follows from (76) that this choice is well defined and satisfies the assumption of Proposition 3 for k sufficiently large. Since the conclusion of this proposition asserts that the method eventually reduces to Newton's method restricted on the subspace T a superlinear convergence rate result follows. This type of argument can be used to construct a number of Newton-like and quasi-Newton methods and prove corresponding convergence and rate of convergence results. We state one of the simplest such results regarding a Newton-like algorithm which is well suited for problems where f is strictly convex and twice differentiable. Its proof follows simply from the preceding discussion and standard results on the unconstrained form of Newton's method so it is left to the reader.

PROPOSITION 4. *Let f be convex and twice continuously differentiable. Assume that problem* (1) *has a unique optimal solution x^* satisfying Assumption* (C), *and that there exist positive scalars m_1, m_2 such that*

$$m_1|z|^2 \leqq z'\nabla^2 f(x) z \leqq m_2|z|^2, \qquad \forall z \in \{x \mid f(x) \leqq f(x_0)\}.$$

Assume also that in the algorithm (32)–(37), *the matrix D_k is given by*

$$D_k = H_k^{-1},$$

where H_k is the matrix with elements H_k^{ij} given by

$$H_k^{ij} = \begin{cases} 0 & \text{if } i \neq j, \text{ and either } i \in I_k^+ \text{ or } j \in I_k^+, \\ \dfrac{\partial^2 f(x_k)}{\partial x^i \partial x^j} & \text{otherwise.} \end{cases}$$

Then the sequence $\{x_k\}$ generated by iteration (35) *converges to x^* and the rate of convergence of $\{|x_k - x^*|\}$ is superlinear (at least quadratic if $\nabla^2 f$ is Lipschitz continuous in a neighborhood of x^*).*

Note that by making use of the result of Proposition 3 it follows that *when f is a positive definite quadratic function, the algorithm of Proposition 4 solves problem* (1) *in a finite number of iterations* provided the unique solution x^* satisfies Assumption (C).

The algorithm of Proposition 4 also has the property that, for all k sufficiently large, the initial unity stepsize will be accepted by the Armijo rule. Our computational experience suggests that the unity stepsize is also acceptable for the great majority

of iterations even before the binding constraints at the solution are identified. We did observe however some cases where it was necessary to reduce the initial unity stepsize several times before a sufficient reduction in the objective function value was effected. The most typical situation where such a phenomenon can occur is when the scalar $\hat{\gamma}_k$ defined by

$$\hat{\gamma}_k = \min\{1, \hat{\alpha}_k\}, \qquad \hat{\alpha}_k = \sup\{\alpha \mid x_k^i - \alpha p_k^i \geqq 0,\ x_k^i > 0,\ i \notin I_k^+\}$$

is small relative to unity. Under these circumstances some nonbinding constraint, that was not taken into account when forming the index set I_k^+, is encountered after a small movement along the arc $\{x_k(\alpha) \mid \alpha > 0\}$. As a result it may occur that the objective function value increases as α is increased from $\hat{\gamma}_k$. A reasonable heuristic device to avoid a large number of function evaluations in such cases is to modify the line search so that if at any iteration a fixed number r of trial stepsizes $1, \beta, \cdots, \beta^{r-1}$ fail to pass the Armijo rule test then $\hat{\gamma}_k$ is computed and used as the next trial stepsize.

There is another (infrequent) situation where a unity initial stepsize may be inappropriate when far from convergence, and the Armijo rule may need a large number of stepsize reductions before determining an acceptable stepsize. This situation can arise when the sets of indices $\{i \mid x_k^i = 0,\ i \notin I_k^+\}$ and $\{i \mid x_k^i = 0,\ p_k^i < 0,\ i \notin I_k^+\}$ are not equal, and as a result the initial direction of motion along the arc $\{x_k(\alpha) \mid \alpha \geqq 0\}$ is not a Newton direction along any subspace. A difficulty of this type can be easily detected and can be typically corrected by combining the Armijo rule with some form of a line minimization rule.

Extension to upper and lower bounds. The algorithm (32)–(37) described so far in this section can be easily extended to handle problems of the form

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } b_1 \leqq x \leqq b_2, \end{aligned} \tag{80}$$

where b_1 and b_2 are given vectors of lower and upper bounds with $b_1 \leqq b_2$. The set I_k^+ is replaced by

$$I_k^{\#} = \left\{ i \,\middle|\, b_1^i \leqq x_k^i \leqq b_1^i + \varepsilon_k \text{ and } \frac{\partial f(x_k)}{\partial x^i} > 0 \text{ or } b_2^i - \varepsilon_k \leqq x_k^i \leqq b_2^i \text{ and } \frac{\partial f(x_k)}{\partial x^i} < 0 \right\}, \tag{81}$$

and the definition of $x_k(\alpha)$ is changed to

$$x_k(\alpha) = [x_k - \alpha D_k \nabla f(x_k)]^{\#}, \tag{82}$$

where for all $z \in R^n$ we denote by $[z]^{\#}$ the vector with coordinates

$$[z]^{\#i} = \begin{cases} b_2^i & \text{if } b_2^i \leqq z^i, \\ z^i & \text{if } b_1^i < z^i < b_2^i, \\ b_1^i & \text{if } z^i \leqq b_1^i. \end{cases} \tag{83}$$

The scalar ε_k is given by $\varepsilon_k = \min\{\varepsilon, |x_k - [x_k - M\nabla f(x_k)]^{\#}|\}$. The matrix D_k is positive definite and diagonal with respect to $I_k^{\#}$, and M is a fixed diagonal positive definite matrix. The iteration is given by

$$x_{k+1} = x_k(\alpha_k), \tag{84}$$

where α_k is chosen by the Armijo rule (36), (37), with $[x_k^i - x_k^i(\beta^m)]^+$ replaced by $[x_k^i - x_k^i(\beta^m)]^{\#}$.

The preceding algorithm also makes sense if some of the upper bounds b_2^i equal $+\infty$ and some of the lower bounds b_1^i equal $-\infty$. This covers the case where only some of the variables x^i are simply constrained by upper and/or lower bounds.

3. Extensions to general linear constraints. In this section we discuss briefly how the algorithms of the previous section can form the basis for constructing methods for solving the problem

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } b_1 \leqq Ax \leqq b_2, \end{aligned} \tag{85}$$

where $f: R^n \to R$ is a continuously differentiable function, A is an $m \times n$ matrix and b_1, b_2 are given vectors. We denote by a_j', $j = 1, \cdots, m$ the rows of A and by $b_{1,j}$, $b_{2,j}$, $j = 1, \cdots, m$ the coordinates of b_1 and b_2, respectively, so the constraint set is represented by the m inequality constraints

$$b_{1,j} \leqq a_j'x \leqq b_{2,j}, \qquad j = 1, \cdots, m. \tag{86}$$

By slight abuse of standard mathematical notation *we allow the possibilities* $b_{1,j} = -\infty$ *and* $b_{2,j} = +\infty$. In this way each of the inequalities (86) may represent a two-sided inequality constraint ($-\infty < b_{1,j} \leqq b_{2,j} < +\infty$), a one-sided inequality constraint ($-\infty = b_{1,j} < b_{2,j} < +\infty$ or $-\infty < b_{1,j} < b_{2,j} = +\infty$) or no constraint at all ($b_{1,j} = -\infty$, $b_{2,j} = +\infty$). When $b_{1,j} = b_{2,j}$ then (86) represents an equality constraint. *We assume that problem* (85) *has at least one feasible solution.* We denote for every feasible x

$$B(x) = \{j \mid b_{1,j} = a_j'x \text{ or } a_j'x = b_{2,j}\}. \tag{87}$$

We assume that for every feasible x *the set of vectors*

$$\{a_j \mid j \in B(x)\} \tag{88}$$

is linearly independent. This is essentially a nondegeneracy assumption. It can be dispensed with at the expense of technical complications which are beyond the scope of the paper. In order to simplify the statement of the algorithm that follows *we assume that the set of inequality constraints* (86) *includes the trivial inequalities*

$$-\infty \leqq x^i \leqq +\infty, \qquad i = 1, \cdots, n, \tag{89}$$

for which a_j is a unit coordinate vector and $b_{1,j} = -\infty$, $b_{2,j} = +\infty$. The value of this somewhat unorthodox device will become apparent shortly.

In the algorithm to be described, given a feasible vector x_k obtained at iteration k, we select a subset $B_k \subset \{1, \cdots, m\}$ *containing exactly* n *indices* and satisfying the following two conditions

(a) $B(x_k) \subset B_k$.

(b) The set of vectors $\{a_j \mid j \in B_k\}$ is linearly independent.

Such a choice is always possible since the set of vectors $\{a_j \mid j \in B(x_k)\}$ is linearly independent by earlier assumption, and it is always possible to supplement the set $B(x_k)$ with a suitable subset of indices corresponding to the trivial constraints (89) so as to form a set B_k satisfying (a) and (b) above. However this may not be the only possibility and the manner in which the set B_k is formed is left open at this point. The set

$$X_k = \{x \mid b_{1,j} \leqq a_j'x \leqq b_{2,j},\ j \in B_k\} \tag{90}$$

is referred to as the *active generalized rectangle at iteration* k. It plays a role similar to the one of the manifold of active constraints in manifold suboptimization methods.

By rearranging indices if necessary, we assume without loss of generality that B_k consists of the first n indices, i.e. $B_k = \{1, 2, \cdots n\}$. Then A is written as

$$A = \begin{bmatrix} A_k^+ \\ A_k^- \end{bmatrix},$$

where A_k^+ is the $n \times n$ invertible matrix having a_j', $j \in B_k$, as its rows. We partition similarly the vectors b_1, b_2;

$$b_1 = \begin{bmatrix} b_{1,k}^+ \\ b_{1,k}^- \end{bmatrix}, \qquad b_2 = \begin{bmatrix} b_{2,k}^+ \\ b_{2,k}^- \end{bmatrix}.$$

The idea of the algorithm is to consider at the (k + 1)st iteration the transformation of variables

$$y = A_k^+ x, \tag{91}$$

by means of which the active generalized rectangle X_k of (90) is transformed into the (ordinary) rectangle

$$Y_k = \{y \mid b_{1,k}^+ \leqq y \leqq b_{2,k}^+\}, \tag{92}$$

while problem (85) is transformed into the problem

$$\begin{aligned} &\text{minimize } h_k(y) \triangleq f[(A_k^+)^{-1} y] \\ &\text{subject to } y \in Y_k,\ b_{1,k}^- \leqq A_k^-(A_k^+)^{-1} y \leqq b_{2,k}^-. \end{aligned} \tag{93}$$

Let $y_k = A_k^+ x_k$. By construction we have that the constraints

$$b_{1,k}^- \leqq A_k^-(A_k^+)^{-1} \leqq b_{2,k}^- \tag{94}$$

are not binding at y_k, so we temporarily ignore them and carry out an iteration of the method of the previous section in the space of variables y. It takes the form (cf. (81)–(84))

$$y_{k+1} = y_k(\alpha_k), \tag{95a}$$

where

$$y_k(\alpha) = [y_k - \alpha D_k \nabla h_k(y_k)]^{\#} \quad \forall \alpha \geqq 0; \tag{95b}$$

D_k is a positive definite matrix which is diagonal with respect to the appropriate set of indices, and $[\cdot]^{\#}$ denotes projection on the rectangle Y_k of (92). The stepsize α_k is selected by means of the Armijo-like rule of the previous section subject, however, to the additional restriction that it belongs to the set of stepsizes

$$\{\alpha \mid b_{1,k}^- \leqq A_k^-(A_k^+)^{-1} y_k(\alpha) \leqq b_{2,k}^-\}$$

that do not lead to violation of the nonbinding constraints (94). Since this set contains an interval of the form $[0, \bar{\alpha}]$, where $\bar{\alpha} > 0$, it is clear that the Armijo-like rule will yield a stepsize after a finite number of arithmetic operations. Taking into account the fact that the gradient of the transformed objective function is

$$\nabla h_k(y_k) = [(A_k^+)']^{-1} \nabla f(x_k)$$

and making use of (91) we can finally write iteration (95) in terms of the original variables as

$$x_{k+1} = (A_k^+)^{-1}[A_k^+ x_k - \alpha_k D_k [(A_k^+)']^{-1} \nabla f(x_k)]^{\#}, \tag{96}$$

where $x_{k+1} = (A_k^+)^{-1} y_{k+1}$. The algorithmic process by means of which x_{k+1} is obtained is illustrated in Fig. 3.

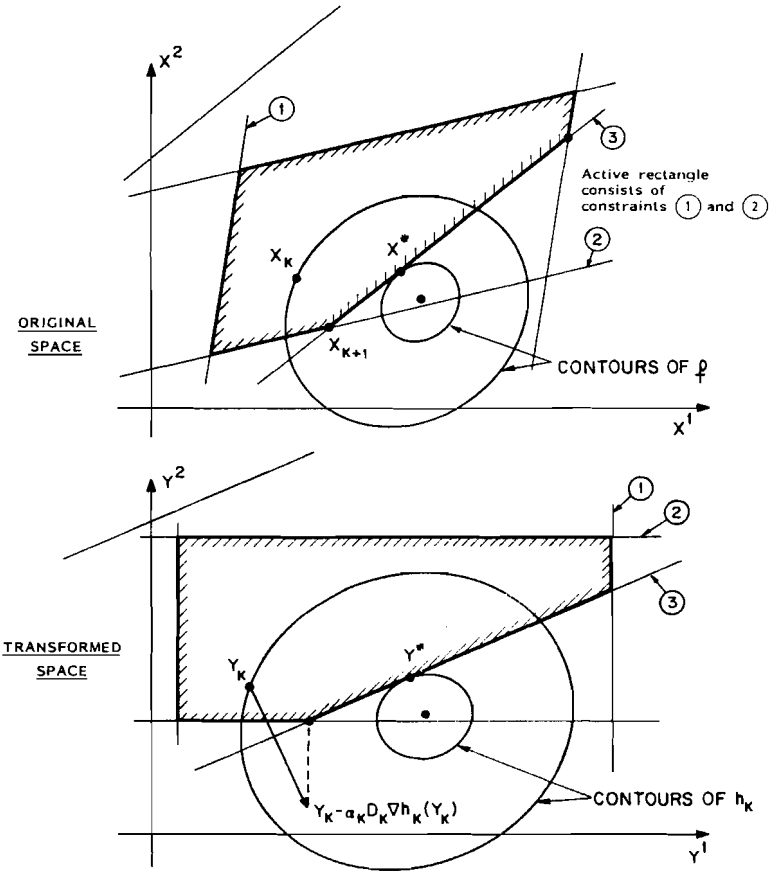

FIG. 3

It may appear that iteration (96) involves excessive computational overhead in view of the presence of the inverse $(A_k^+)^{-1}$. However in many problems with special structure it is possible to compute this inverse very efficiently. For other problems we note that it is possible to organize the algorithm so that most of the indices in the sets B_k and B_{k+1} are common. In fact at each iteration k typically at most one nonbinding inequality not belonging to the current active rectangle B_k will become binding at the next iteration, the exception being the unlikely situation where more than one of the constraints (94) will become simultaneously binding at y_{k+1}. In this case the *matrices A_k^+ and A_{k+1}^+ need only differ by at most one row and as a result the inverse $(A_{k+1}^+)^{-1}$ can be obtained from $(A_k^+)^{-1}$ by the Householder modification rule involving only $O(n^2)$ arithmetic operations* (see Gill and Murray [8, p. 59]). Note also that if a number (say n_k) of the trivial constraints (89) participate in the formation of the active rectangle (90) then the inverse $(A_k^+)^{-1}$ can be formed by matrix inversion of order $(n - n_k)$.

The reader who is familiar with manifold suboptimization methods, as described for example in Gill and Murray [8], will notice a strong similarity between the transformation process involved in these methods and the one employed above. The only essential difference is that in our method we use the active generalized rectangle X_k in place of the manifold of active constraints. The main advantage that algorithm (96) offers over manifold suboptimization alternatives is that *as many as n new*

constraints may become binding in a single iteration while considerable flexibility is afforded in changing the active set of constraints. By contrast, in manifold suboptimization methods, barring exceptional circumstances, at most *one* new constraint will become binding in any single iteration while dropping currently active constraints must be carefully controlled. Thus a *fundamental limitation of these methods is substantially overcome, the capability of attaining superlinear convergence is maintained and there is no need to solve a quadratic programming subproblem at each iteration.*

There are many issues relating to convergence, rate of convergence, active rectangle selection and implementation of the algorithm described in this section but their discussion properly belongs to a separate paper. We provide instead a specific superlinearly convergent Newton-like implementation of algorithm (96) for the case where the constraint set is a simplex.

Example (*minimization on a simplex*). Consider the problem

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \geqq 0, \ \sum_{i=1}^{n} x^i = 1, \end{aligned} \tag{97}$$

where we assume that the function $f: R^n \to R$ is convex, twice continuously differentiable with everywhere positive definite Hessian matrix. Given a feasible vector x_k let

$$\bar{i} = \arg\max \{x_k^i | i = 1, \cdots, n\}.$$

We consider the transformation of variables defined by

$$y^i = x^i \quad \forall\, i \neq \bar{i}, \qquad y^{\bar{i}} = \sum_{i=1}^{n} x^i,$$

thus implicitly forming an active rectangle consisting of the equation $\sum_{i=1}^{n} x_i = 1$ and the inequalities $x^i \geqq 0$, $i \neq \bar{i}$. The inverse transformation is

$$x^i = y^i \quad \forall\, i \neq \bar{i}, \qquad x^{\bar{i}} = y^{\bar{i}} - \sum_{i \neq \bar{i}} y^i.$$

If we write this transformation as $x = T_k y$, where T_k is the appropriate matrix, the problem is transformed into

$$\begin{aligned} &\text{minimize } h_k(y) \triangleq f(T_k y) \\ &\text{subject to } y^i \geqq 0 \ \forall i \neq \bar{i}, \\ &\qquad y^{\bar{i}} = 1, \\ &\qquad y^{\bar{i}} - \sum_{i \neq \bar{i}} y^i \geqq 0. \end{aligned}$$

The last constraint is (by construction) inactive at the point $y_k = T_k x_k$, so it will be ignored in the iteration of the Newton-like method of § 2.

The first and second derivatives of the transformed objective function h_k with respect to the variables y^i at y_k are given by

$$\frac{\partial h_k(y_k)}{\partial y^i} = \frac{\partial f(x_k)}{\partial x^i} - \frac{\partial f(x_k)}{\partial x^{\bar{i}}} \quad \forall\, i \neq \bar{i},$$

$$\frac{\partial h_k(y_k)}{\partial y^{\bar{i}}} = \frac{\partial f(x_k)}{\partial x^{\bar{i}}},$$

$$\frac{\partial^2 h_k(y_k)}{\partial y^i \partial y^j} = \frac{\partial^2 f(x_k)}{\partial x^i \partial x^j} - \frac{\partial^2 f(x_k)}{\partial x^i \partial x^{\bar{\imath}}} - \frac{\partial^2 f(x_k)}{\partial x^j \partial x^{\bar{\imath}}} + \frac{\partial^2 f(x_k)}{(\partial x^{\bar{\imath}})^2} \quad \forall\, i \neq \bar{\imath}, \quad j \neq \bar{\imath},$$

$$\frac{\partial^2 h_k(y_k)}{\partial y^i \partial y^{\bar{\imath}}} = \frac{\partial^2 f(x_k)}{\partial x^i \partial x^{\bar{\imath}}} - \frac{\partial^2 f(x_k)}{(\partial x^{\bar{\imath}2}} \quad \forall\, i \neq \bar{\imath},$$

$$\frac{\partial^2 h_k(y_k)}{(\partial y^{\bar{\imath}})^2} = \frac{\partial^2 f(x_k)}{(\partial x^{\bar{\imath}})^2}.$$

The Newton-like iteration to be performed in the space of variables y is a slight variation of the one of § 2 (cf. Proposition 3) to account for the presence of the constraint $y^{\bar{\imath}} = 1$. It takes the form described below. Let

$$w_k = \left\{ \sum_{i \neq \bar{\imath}} \left(y_k^i - \left[y_k^i - \mu_k^i \frac{\partial h_k(y_k)}{\partial y^i} \right]^+ \right)^2 \right\}^{1/2},$$

where $\mu_k^i = [\partial^2 h_k(y_k)/(\partial y^i)^2]^{-1}$. Let also

$$\varepsilon_k = \min\{\varepsilon, w_k\}, \qquad I_k^+ = \left\{ i \,\middle|\, 0 \leqq y_k^i \leqq \varepsilon_k, \frac{\partial h_k(y_k)}{\partial y^i} > 0 \right\},$$

and form the matrix H_k with elements H_k^{ij} given by

$$H_k^{ij} = \begin{cases} 0 & \text{if } i \neq j \text{ and either } i \in I_k^+ \text{ or } j \in I_k^+, \\ \dfrac{\partial^2 h_k(y_k)}{\partial y^i \partial y^j} & \text{otherwise.} \end{cases}$$

Let

$$p_k = H_k^{-1} \nabla h_k(y_k). \tag{98}$$

Then $y_{k+1} = y_k(\alpha_k)$, where for all $\alpha \geqq 0$

$$y_k^i(\alpha) = [y_k^i - \alpha p_k^i]^+ \quad \forall\, i \neq \bar{\imath}, \qquad y_k^{\bar{\imath}}(\alpha) = 1.$$

The stepsize α_k is given by $\alpha_k = \beta^{m_k}$, where m_k is the first nonnegative integer m such that

$$h_k(y_k) - h_k[y_k(\beta^m)] \geqq \sigma \left\{ \beta^m \sum_{i \in I_k^+} \frac{\partial h_k(y_k)}{\partial y^i} p_k^i + \sum_{i \in I_k^+} \frac{\partial h_k(y_k)}{\partial y^i} [y_k^i - y_k^i(\beta^m)] \right\}$$

and

$$1 - \sum_{i \neq \bar{\imath}} y_k^i(\beta^m) \geqq 0.$$

The vector x_{k+1} is then given by

$$x_{k+1}^i = y_{k+1}^i \quad \forall\, i \neq \bar{\imath}, \qquad x_{k+1}^{\bar{\imath}} = 1 - \sum_{i \neq \bar{\imath}} y_{k+1}^i.$$

Similarly as for Proposition 3, it is easily shown that this algorithm converges superlinearly to the unique (global) minimum of problem (97). The algorithm can be extended trivially to the case where, in addition to the nonnegativity constraints, there is a single equality or inequality constraint, as well as to the case where the constraint set consists of a Cartesian product of simplices. Similar algorithms can be written in

explicit form for problems with a large number of nonnegativity (or upper and lower bound constraints) and a small number of additional equality or inequality constraints. Newton-like algorithms of this type are particularly effective when the problem has special structure that facilitates the solution of the linear system of equations involved in implementing the basic iteration (cf. (98)).

4. Application in discrete-time optimal control—computational results. The algorithms of the paper are particularly well suited for discrete-time optimal control problems involving a discrete time system of the form

$$x_{i+1} = f_i(x_i, u_i), \qquad i = 0, \cdots, N-1, \tag{99}$$

a cost functional of the form

$$G(x_N) + \sum_{i=0}^{N-1} g_i(x_i, u_i) \tag{100}$$

and simple constraints on the control vectors of the form

$$\underline{b}_i \leqq u_i \leqq \bar{b}_i, \qquad i = 0, \cdots, N-1.$$

We assume that the functions $f_i : R^{n+m} \to R^n$, $g_i : R^{n+m} \to R$ and $G : R^n \to R$ are twice continuously differentiable and N is a positive integer. Problems of this type are discussed for example in Varaiya [17], Polak [18], and Cannon, Cullum and Polak [20]. They are often characterized by large dimension, particularly when they arise from discretization of continuous-time optimal control and calculus of variations problems.

Each state vector x_i can be uniquely represented in terms of the control sequence $u = \{u_0, \cdots, u_{N-1}\}$ via the system equation (99) in the form

$$x_i = \phi_i(u), \qquad i = 1, \cdots, N,$$

where ϕ_i are the appropriate functions. The problem is then equivalent to

$$\begin{aligned} &\text{minimize } J(u) = G[\phi_N(u)] + \sum_{i=0}^{N-1} g_i[\phi_i(u), u_i] \\ &\text{subject to } \underline{b}_i \leqq u_i \leqq \bar{b}_i, \qquad i = 0, \cdots, N-1. \end{aligned} \tag{101}$$

It is well known (see Mitter [21], Polak [18]) that the unconstrained Newton direction $-[\nabla^2 J(u)]^{-1} \nabla J(u)$ for this problem can be efficiently computed by means of the Riccati equation. An algorithm such as the one of Proposition 4 can also be similarly implemented via the Riccati equation. At each iteration k we first determine the set of indices $I_k^{\#}$ [cf. (81)]. We then compute the Newton direction with respect to the control vector coordinates corresponding to indices $i \notin I_k^{\#}$ via the Riccati equation, while we compute the (diagonally) scaled steepest descent direction for the remaining coordinates corresponding to indices $i \in I_k^{\#}$. The overall algorithm is thus very similar to the one used for the corresponding unconstrained problem. It is well suited for large scale linear-quadratic problems with simple control constraints for which pivoting methods are apparently very cumbersome and inefficient. Our computational example is of this type.

Consider the two-dimensional linear system

$$\begin{bmatrix} x_{i+1,1} \\ x_{i+1,2} \end{bmatrix} = \begin{bmatrix} 1 & s \\ -s & 1 \end{bmatrix} \begin{bmatrix} x_{i,1} \\ x_{i,2} \end{bmatrix} + \begin{bmatrix} 0 \\ s \end{bmatrix} u_i, \qquad i = 0, 1, \cdots, N-1. \tag{102}$$

The initial state $x_0 = (x_{0,1}, x_{0,2})$ is given and the control constraints are

$$-1 \leqq u_i \leqq 1, \qquad i = 0, 1, \cdots, N-1. \tag{103}$$

The problem is to minimize

$$J(u) = \frac{s}{2} \sum_{i=0}^{N-1} (x'_{i+1} Q x_{i+1} + R u_i^2), \tag{104}$$

where the matrix Q and the scalar R are given by

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}, \qquad R = 6. \tag{105}$$

This problem arises from discretization of the continuous-time problem of minimizing

$$\frac{1}{2} \int_0^T [x(t)' Q x(t) + R u(t)^2] \, dt \tag{106}$$

subject to

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \tag{107}$$

and

$$-1 \leqq u(t) \leqq 1, \qquad t \in [0, T]. \tag{108}$$

If the interval $[0, T]$ is discretized into N intervals of length

$$s = \frac{T}{N} \tag{109}$$

and the approximation

$$\dot{x}(t) \simeq \frac{1}{5}(x_{i+1} - x_i), \qquad t \in [is, (i+1)s] \tag{110}$$

is used, then problem (102)–(105) is a discretized version of problem (106)–(110).

We show in Table 1 for a variety of values of N and s the number of iterations required by the method of Proposition 4 to obtain the exact solution for two initial states $x_0 = (15, 5)$ and $x_0 = (5, -10)$ and two initial control trajectories $u_i^0 \equiv 0$ and $u_i^0 \equiv 1$. In all runs we chose $\varepsilon = 0.01$, $\beta = 0.5$ and $\sigma = 10^{-4}$. All computations were performed in double precision and this was found essential for large values of N. The results demonstrate the ability of the method to identify the set of binding constraints in very few iterations. It is worth noting that while the table gives results for N only up to 10,000, an incomplete set of experiments was run with $N = 25{,}000$, and a very similar performance was observed for the method.

TABLE 1

x_0	N	s	# of binding constraints at solution	# of iterations	
				$u_i^0 \equiv 0$	$u_i^0 \equiv 1$
(15, 5)	100	0.002	0	1	1
		0.01	18	3	2
		0.04	78	3	3
		0.1	91	4	3
		0.5	100	5	5
	1,000	0.0002	0	1	1
		0.001	183	3	2
		0.004	770	4	3
		0.01	890	4	3
		0.05	705	23	16
	10,000	0.00002	0	1	1
		0.0001	1834	3	3
		0.0004	7693	5	4
		0.001	8861	6	4
		0.005	4261	10	18
(5, −10)	100	0.002	0	1	1
		0.01	48	2	1
		0.04	73	4	3
		0.1	87	5	4
		0.5	100	7	5
	1,000	0.0002	0	1	1
		0.001	478	3	1
		0.004	684	4	3
		0.01	765	5	4
		0.05	370	9	18
	10,000	0.00002	0	1	1
		0.0001	4772	3	2
		0.0004	6802	5	7
		0.001	7595	6	5
		0.005	2591	12	17

REFERENCES

[1] A. A. GOLDSTEIN, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc., 70 (1964), pp. 709–710.

[2] E. S. LEVITIN AND B. T. POLJAK, *Constrained minimization problems*, U.S.S.R. Comput. Math. Math. Phys., 6 (1966), pp. 1–50.

[3] G. P. MCCORMICK, *Anti-zig-zagging by bending*, Management Science, 15 (1969), pp. 315–319.

[4] D. P. BERTSEKAS, *On the Goldstein–Levitin–Poljak gradient projection method*, IEEE Trans. Automat. Control, AC-21 (1976), pp. 174–184.

[5] J. C. DUNN, *Newton's method and the Goldstein step-length rule for constrained minimization problems*, this Journal, 6 (1980), pp. 659–674.

[6] U. M. GARCIA-PALOMARES AND O. L. MANGASARIAN, *Superlinearly convergent quasi-Newton algorithms for nonlinearly constrained optimization problems*, Math. Prog., 11 (1976), pp. 1–13.

[7] J. B. ROSEN, *The gradient projection method for nonlinear programming, Part* I: *linear constraints*, J. Soc. Ind. Appl. Math., 8 (1960), pp. 181–217.

[8] P. E. GILL AND M. MURRAY, eds., *Numerical Methods for Constrained Optimization*, Academic Press, New York, Chs. 2, 3.

[9] D. GOLDFARB, *Extension of Davidon's variable metric algorithm to maximization under linear inequality and equality constraints*, SIAM J. Applied Math., 17 (1969), pp. 739–764.

[10] D. G. LUENBERGER, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973, Ch. 11.

[11] M. L. LENARD, *A computational study of active set strategies in nonlinear programming with linear constraints*, Math. Prog. 16 (1979), pp. 81–97.

[12] G. P. MCCORMICK, *The variable reduction method for nonlinear programming*, Management Science, 17 (1970), pp. 146–160.

[13] R. K. BRAYTON AND J. CULLUM, *An algorithm for minimizing a differentiable function subject to box constraints and errors*, J. Opt. Th. Appl., 29 (1979), pp. 521–558.

[14] W. MURRAY, *Second derivative methods*, Numerical Methods for Unconstrained Optimization, W. Murray, ed., Academic Press, New York, 1972, pp. 57–71.

[15] R. FLETCHER AND T. L. FREEMAN, *A modified Newton method for minimization*, J. Opt. Th. Appl., 23 (1977), pp. 357–372.

[16] J. J. MORÉ AND D. C. SORENSEN, *On the use of directions of negative curvature in a modified Newton method*, Math. Prog., 16 (1979), pp. 1–20.

[17] P. P. VARAIYA, *Notes on Optimization*, Van Nostrand–Reinhold, New York, 1972.

[18] E. POLAK, *Computational Methods in Optimization: A Unified Approach*, Academic Press, New York, 1971.

[19] D. P. BERTSEKAS, *Constrained Optimization and Lagrange Multiplier Methods*, Academic Press, New York, 1981 (to appear).

[20] M. D. CANNON, C. D. CULLUM AND E. POLAK, *Theory of Optimal Control and Mathematical Programming*, McGraw-Hill, New York, 1970.

[21] S. K. MITTER, *Successive approximation methods for the solution of optimal control problems*, Automatica, 3 (1966), pp. 135–149.

[22] J. C. DUNN, *Global and asymptotic convergence rate estimates for a class of projected gradient processes*, this Journal, 19 (1981), pp. 368–400.