Mathematical Programming 2 (1972) 65–106. North-Holland Publishing Company

AN EFFICIENT ALGORITHM FOR MINIMIZING BARRIER AND PENALTY FUNCTIONS *

Leon S. LASDON
Case Western Reserve University, Cleveland, Ohio, U.S.A.

Received 10 January 1971
Revised manuscript received 15 July 1971

Constrained minimization is often done via interior penalty functions. Such functions can be very difficult to minimize using existing algorithms. In this paper, a new algorithm is described which is specially constructed to deal with such functions. It generates search directions by linearizing the objective and constraints about the current (interior) point, substituting these linearizations into the penalty function, and minimizing the result. Properties of the algorithm are derived, an efficient method for solving the direction finding problem is suggested, and computational results are presented. Preliminary results are also given on an extension to quasi-barrier and exterior penalty functions.

1. Introduction

Penalty and barrier methods (e.g., exterior and interior penalty methods) for solving nonlinear programs are now widely used [1]. These solve a nonlinear constrained optimization problem by solving a sequence of unconstrained problems. Their popularity is due to their simplicity – they enable any unconstrained minimizer, with slight modification, to solve a constrained problem, and to their reliability-loosely speaking, the unconstrained minima found converge to a solution of the constrained problem. However, the unconstrained problems can become infinitely ill-conditioned as the penalty parameter tends to its limiting value [2–3]. That is, the ratio of largest to smallest eigenvalue of the

* This document may be reproduced in whole or in part for any non-commercial purpose of the United States Government. Its preparation was supported in part by funds allocated to Case Western Reserve University under contract DAHC 19–68–C–0007 (Project Themis) with the U.S. Army Research Office, Durham Army Materiel Command.

Hessian matrix of the penalty function, evaluated at the unconstrained minimum, can become infinite. Hence, efficient unconstrained minimizers are needed.

Powerful general purpose methods exist, e.g., a modified Newton method and algorithms of the Davidon type [4]. However, these all have certain deficiencies. Newtons method requires second derivatives, and coding these can be tedious, sometimes practically impossible, for complex problems. Davidon and other conjugate direction methods require only first derivatives, but usually take more iterations than the Newton procedures [5] (although not necessarily more computing time), and often require that an accurate one-dimensional search be performed [6]. This latter requirement is particularly difficult to meet when minimizing penalty or barrier functions [7]. Moreover, being general purpose procedures, none of these take advantage of the special structure of penalty and barrier functions. Focusing for the moment on barrier functions, these are hard to minimize because they contain terms which approach infinity as the argument approaches zero. Newton and conjugate direction methods use a quadratic to approximate such functions, and quadratics do not approach infinity at any finite point. Because of this, the region over which a given quadratic adequately represents such functions can be rather small. There is intuitive appeal to using instead an approximating function which also approaches infinity, and does so roughly where the barrier function does. Such a function could adequately approximate the barrier function over a large region, so that its minimum would be close to the barrier minimum. Finding this minimum and doing a one-dimensional search in its direction forms one step of an iterative algorithm.

In this paper we propose such an approximating function and develop an algorithm based on it. By exploiting special structural features of penalty and barrier functions – some very strong monotonicity properties – we prove that the search directions constructed are always directions of descent, and that the algorithm converges in the limit. For problems with convex objective and concave constraints, accurate estimates of the Lagrange multipliers and a lower bound on the penalty function minimum are readily available. The problem of minimizing the approximating function can be formulated so as not to become infinitely ill-conditioned and has a great deal of structure. Some encouraging computational results are also given. Finally, extensions to quasi-barrier and exterior penalty functions are outlined.

2. The $\tilde{P}$-algorithm for barrier functions

The nonlinear program whose solution is desired is

Problem NLP.

$$\begin{array}{ll} \text{Minimize} & f(x) \\ \text{Subject to} & g_i(x) \geq 0, \quad i = 1, \ldots, m \end{array}$$

The feasible region is

$$F = \{x \mid g_i(x) \geq 0, \quad i = 1, \ldots, m\} . \tag{2.1}$$

We assume the following

Assumption 1. f and all g_i have continuous * first partial derivatives at all points in F.

In this section, we consider solving NLP by a barrier algorithm. Since these move toward a solution from points where all g_i are positive, we must assume

Assumption 2.

$$F^{\mathrm{o}} = \{x \mid g_i(x) > 0, \quad i = 1, \ldots, m\} \neq \emptyset . \tag{2.2}$$

Central to the algorithm is a barrier function B, defined for positive real numbers, having the following properties [8]

1. $B(z)$ is continuous for $z > 0$
2. $\lim_{z \to 0^+} B(z) = +\infty$.

Using B, we define a P-function

$$P(x, r) = f(x) + r \sum_{i=1}^{m} B(g_i(x)) \tag{2.3}$$

where r is a positive scalar, and a

* The continuity requirement is needed only in the convergence proof.

P-problem

$$\begin{array}{ll} \text{minimize} & P(x, r) \\ \text{subject to} & x \in F . \end{array}$$

It is shown in [8, 9] that, if f and all g_i are continuous, $F^\circ \neq \emptyset$, and F is compact, then the P-problem has an optimal solution, and any such solution is in F°. A barrier algorithm for solving NLP proceeds by choosing a positive, decreasing sequence of values $\{r_k\}$ tending to zero and solving the corresponding sequence of P-problems. Convergence of this algorithm to an optimal solution of NLP can be proved under very mild additional assumptions (see [8, 9]).

Almost all the computational effort in applying a barrier algorithm is expended in solving the P-problems. We now propose a new algorithm for this purpose. Let $x^* \in F^\circ$ be the current value of x, and define

$$a_0 = f(x^*), \qquad b_0 = \nabla f(x^*) \tag{2.4}$$

$$a_i = g_i(x^*), \qquad b_i = \nabla g_i(x^*), \qquad i = 1, ..., m \tag{2.5}$$

using this data, we define an approximating function for P at the point x^* as

$$\widetilde{P}(x; x^*, r) = a_0 + b_0^t(x-x^*) + r \sum_{i=1}^{m} B(a_i + b_i^t(x-x^*)) . \tag{2.6}$$

That is, we form $\widetilde{P}$ by taking the P-function and replacing f and each g_i by their linearization about x^*.

Note that

$$\widetilde{P}(x^*; x^*, r) = P(x^*, r) . \tag{2.7}$$

Further, assuming $B(z)$ is differentiable for positive z

$$\nabla_x \widetilde{P}(x; x^*, r) = b_0 + r \sum_{i=1}^{m} B'(a_i + b_i^t(x-x^*))b_i \tag{2.8}$$

since

$$\nabla P(x^*, r) = b_0 + r \sum_{i=1}^{m} B'(a_i)b_i \tag{2.9}$$

we see that

$$\nabla_x \widetilde{P}(x^*; x^*, r) = \nabla P(x^*, r) . \tag{2.10}$$

Hence $\widetilde{P}$ and P have the same values and gradients at the point x^*.
The domain of definition of $\widetilde{P}$ is the interior of the set

$$\mathrm{L}F(x^*) = \{x \mid a_i + b_i^t(x - x^*) \geq 0, \quad i = 1, ..., m\} . \tag{2.11}$$

$\widetilde{P}$ goes to $+\infty$ on the boundary of LF. Of course, one of the distinguishing features of P is that it goes to $+\infty$ on the boundary of F, and it is intuitively appealing to approximate P with a function having this same feature, especially since LF is a good local approximation to F for x^* near the boundary of F. Of course, $\widetilde{P}$ is a good local approximation to P for any $x^* \in F^o$ since, for points near x^*, f and the g_i are approximately equal to their linearizations.

We use $\widetilde{P}$ to construct an iterative algorithm. Let

$$s = x - x^* \tag{2.12}$$

and define

$$\mathrm{L}S(x^*) = \{s \mid a_i + b_i^t s \geq 0, \quad i = 1, ..., m\} . \tag{2.13}$$

Given a point x^*, we determine a search direction s^* by solving the following direction finding problem:

Problem D$F(x^*)$
minimize

$$\widetilde{P}(s; x^*, r) = a_0 + b_0^t s + r \sum_{i=1}^{m} B(a_i + b_i^t s) \tag{2.14}$$

subject to

$$a_i + b_i^t s \geq 0, \quad i = 1, ..., m \tag{2.15}$$

and the normalization constraint

$$N(s) \leq \delta, \quad \delta > 0 \tag{2.16}$$

where $N(s)$ is any norm for E^n. Since all norms are convex functions, (2.16) defines a convex set.

The normalization constraint is needed because $\widetilde{P}$ may not be a good approximation to P for "large" s. Note that $DF(x^*)$ is feasible, since $s = 0$ satisfies (2.15) – (2.16). Hence $DF(x^*)$ has an optimal solution for any point $x^* \in F^o$, and all solutions satisfy (2.15) strictly.

Having found a direction s^*, we choose a successor point to x^* by solving the one dimensional minimization problem

$$\text{minimize} \qquad P(x^* + \alpha\, s^*, r)$$

subject to the conditions $\alpha \geq 0$ and $x^* + \alpha\, s^* \in F$. The process is iterated. Hence the proposed algorithm is

P-algorithm for barrier functions

0. Start at a point $x_0 \in F^o$. Set $i = 0$.
1. Solve $DF(x_i)$, yielding a solution s_i.
2. Choose $\alpha = \alpha_i$ by minimizing $P(x_i + \alpha s_i, r)$ subject to $x_i + \alpha s_i \in F$ and $\alpha \geq 0$.
3. Set $x_{i+1} = x_i + \alpha_i$, replace i by $i + 1$ and return to step 1.

A variety of termination criteria may be used, usually based on the behavior of the sequences $\{\nabla P(x_i)\}$ or $\{P(x_i)\}$.

We note that Marquardt's method for nonlinear least squares problems [23] uses the same ideas as outlined above. Substituting linearizations into a sum of squares yields a quadratic approximating function. This is minimized within a spherical neighborhood of the current point, and the step size α_i is regulated by varying the radius of the sphere. As shown by Bard [24], Marquardt's method is one of the most efficient for least squares problems.

In order to endow this algorithm with some desirable properties, we make additional assumptions concerning the barrier function B. These are

Assumption 3. For all $z > 0$, B is differentiable, strictly convex, and monotone decreasing.
We note that this implies that $B'(z) = \mathrm{d}B/\mathrm{d}z$ is monotone increasing for $z > 0$, a property which is used later.

The conditions of assumption 3 are satisfied by all commonly used barrier functions [8], [9], e.g., by $B(z) = 1/z$ and $B(z) = -\ln(z)$. With

small modification, they are also satisfied by all commonly used penalty and quasi-barrier functions, as we discuss later.
Under assumption 3, we have

Theorem 1. $\widetilde{P}(s; x^*, r)$ is convex in s over $LS(x^*)$ for any $x^* \in F^o$, and any $r > 0$. Hence any local solution of $DF(x^*)$ is global.
The proof is immediate from the fact that the composition of a convex function and a linear function is convex. Since N is a convex function, the feasible region of $DF(x^*)$ is convex, and the second statement of the theorem follows.

The following result shows that $DF(x^*)$ can produce zero directions if and only if x^* is a stationary point of P. For f convex and the g_i concave, this means that zero directions are produced if and only if x^* minimizes P

Theorem 2. $s^* = 0$ solves $DF(x^*)$ if and only if $\nabla P(x^*, r) = 0$.

Proof. Since $\widetilde{P}$ is convex, 0 solves $DF(x^*)$ if and only if

$$\nabla \widetilde{P}(0; x^*, r) = 0 .$$

But, by (2.8) and (2.9)

$$\nabla \widetilde{P}(0; x^*, r) = \nabla P(x^*, r) . \tag{2.17}$$

An important property of $DF(x^*)$ is that it always produces directions of descent if x^* is not a stationary point of P. This property seems essential in algorithms which use derivatives, and is used in the convergence proof.

Theorem 3. If

a) $\nabla P(x^*, r) \neq 0$.

b) The barrier function B satisfies assumption 3.

c) s^* solves $DF(x^*)$.

Then

$$\nabla P^t(x^*, r) s^* < 0 .$$

Before beginning the main proof, some properties of $N(s)$ must be established. Since some norms, e.g. L_1 and L_∞, are not differentiable at the origin, we must invoke the theory of subgradients * and directional derivatives of convex functions [10]. We use the notation $\partial N(s)$ for the set of subgradients of N at s, dom f for the effective domain of a function f, and $DF(x; d)$ for the (one-sided) directional derivative of a function f at x in the direction d. Since the subgradient theory deals with functions convex over all of E^n, we alter the definition of $\widetilde{P}(s; x^*, r)$ so that it equals $+\infty$ for points not in $\mathrm{L}S^{\mathrm{o}}(x^*)$. $\widetilde{P}$ is then convex over E^n with effective domain $\mathrm{L}S^{\mathrm{o}}(x^*)$.

Lemma 1. Let N be any norm for E^n, s any vector, and $y \in \partial N(s)$. Then

$$y^t s = N(s) . \tag{2.18}$$

Proof. Since N is finite for all $s \in E^n$, it is a proper convex function. Hence, for all $s \in E^n$, $\partial N(s) \neq \emptyset$. By definition, $y \in \partial N(s)$ if and only if

$$N(z) \geq N(s) + y^t(z - s) \qquad \text{for all } z \in E^n . \tag{2.19}$$

Let

$$z = \alpha s, \ \ \alpha > 0 .$$

Since N is a norm

$$N(\alpha s) = \alpha N(s)$$

so (2.19) becomes

$$(\alpha - 1)N(s) \geq (\alpha - 1) y^t s , \ \ \text{for all } \alpha > 0 .$$

If $\alpha > 1$, we can divide the above inequality by $(\alpha - 1)$ yielding

$$N(s) \geq y^t s . \tag{2.20}$$

If $\alpha < 1$, dividing by $\alpha - 1$ changes the sense of the inequality, so

$$N(s) \leq y^t s . \tag{2.21}$$

* y is a subgradient of a function f at a point x if $f(z) \geq f(x) + y^t(z-x)$ for all $z \in E^n$.

Relations (2.20) and (2.21) imply

$$N(s) = y^t s$$

which proves lemma 1.

Lemma 2. Let f_1 and f_2 be proper convex functions with f_1 differentiable over dom f_1, and define

$$f = f_1 + f_2$$

where dom f = dom $f_1 \subseteq$ dom f_2. Let $x \in$ dom f. Then, if $y \in \partial f(x)$, there exists $y_2 \in \partial f_2(x)$ such that

$$y = \nabla f_1(x) + y_2 \; .$$

Proof. We must show that $y - \nabla f_1(x) \in \partial f_2(x)$. By theorem 10 of ref. [10], this is true if and only if

$$\mathrm{D}f_2(x; d) \geq (y - \nabla f_1(x))^t d, \quad \text{for all } d \in E^n \tag{2.22}$$

Since $y \in \partial f(x)$

$$\mathrm{D}f(x; d) \geq y^t d \,, \qquad \text{for all } d \in E^n \; . \tag{2.23}$$

By definition of f

$$\mathrm{D}f(x; d) = \mathrm{D}f_1(x; d) + \mathrm{D}f_2(x; d)$$

$$= \nabla f_1^t(x) d + \mathrm{D}f_2(x; d) \; . \tag{2.24}$$

Using (2.24) in (2.23)

$$\nabla f_1^t(x) d + \mathrm{D}f_2(x; d) \geq y^t d \,, \qquad \text{for all } d \in E^n$$

or

$$\mathrm{D}f_2(x; d) \geq y^t d - \nabla f_1^t(x) d \,, \qquad \text{for all } d \in E^n \; .$$

But this is (2.22), so the lemma is proved.

Proof of theorem 3. By the properties of B

$$a_i + b_i^t s^* > 0, \qquad i = 1, \ldots, m$$

so the only constraint which can be binding in $DF(x^*)$ is the normalization condition. Since $\widetilde{P}$ and the norm function N are convex and the normalization constraint can be satisfied strictly, we may use the saddle point theorem of Karlin [11]. By this result, there exists a multiplier $\lambda^* \geq 0$ such that the Lagrangian function

$$L(s, \lambda; x^*, r) = \widetilde{P}(s; x^*, r) + \lambda(N(s) - \delta) \tag{2.25}$$

has a saddle point at (s^*, λ^*). Necessary and sufficient conditions for this are

1. s^* minimizes $L(s, \lambda^*; x^*, r)$
2. $\lambda^*(N(s^*) - \delta) = 0$
3. $N(s^*) \leq \delta$.

Since $L(s, \lambda^*; x^*, r)$ is proper convex, condition (1) above holds if and only if * $0 \in \partial L(s^*, \lambda^*; x^*, r)$. The function $\widetilde{P}$ is differentiable at s^*, so lemma 2 implies that there is a $y \in \partial N(s^*)$ such that

$$\nabla \widetilde{P}(s^*; x^*, r) + \lambda^* y = 0$$

or, using the expression in (2.8) for $\nabla \widetilde{P}$

$$b_0 + r \sum_{i=1}^{m} B'(a_i + b_i^t s^*) b_i + \lambda^* y = 0 . \tag{2.26}$$

Using (2.9)

$$\nabla P^t(x^*, r) s^* = b_0^t s^* + r \sum_{i=1}^{m} B'(a_i) b_i^t s^* . \tag{2.27}$$

Taking the scalar product of (2.26) with s^* yields

$$b_0^t s^* = -r \sum_{i=1}^{m} B'(a_i + b_i^t s^*) b_i^t s^* - \lambda^* y^t s^* .$$

* See theorem 4, ref. [10].

Substituting the above into (2.27) gives

$$\nabla P^t(x^*, r)s^* = r \sum_{i=1}^{m} (B'(a_i) - B'(a_i + b_i^t s^*))b_i^t s^* - \lambda^* y^t s^* . \tag{2.28}$$

By lemma 1

$$y^t s^* = N(s^*) \geq 0 .$$

Thus, since $\lambda^* \geq 0$, the last term on the right of (2.28) is nonpositive. We now show that the sum in (2.28) must be negative. Consider the term

$$[B'(a_i) - B'(a_i + b_i^t s^*)]\ b_i^t s^* .$$

If $b_i^t s^* < 0$, since B' is monotone increasing, the bracketed term is positive, and the product is negative. The term is also negative if $b_i^t s^* > 0$. Hence the sum in (2.28) is negative if at least one $b_i^t s^* \neq 0$, and the theorem is proved for this case. Consider now the situation where

$$b_i^t s^* = 0, \quad i = 1, ..., m . \tag{2.29}$$

Then, by (2.27)

$$\nabla P^t(x^*, r)\, s^* = b_0^t s^* \tag{2.30}$$

and

$$\widetilde{P}(s^*; x^*, r) = a_0 + r \sum_{i=1}^{m} B(a_i) + b_0^t s^*$$

$$= \widetilde{P}(0, x^*, r) + b_0^t s^* .$$

If $b_0^t s^* > 0$, then $\widetilde{P}$ could be reduced by setting $s^* = 0$, which contradicts the optimality of s^*. Hence $b_0^t s^* \leq 0$. If $b_0^t s^* = 0$, then $s^* = 0$ is also optimal for DF(x^*). By theorem (2), this contradicts the assumption that $\nabla P(x^*, r) \neq 0$.
Hence

$$b_0^t s^* < 0 .$$

By (2.30), this proves the theorem.

3. Special properties for convex programs

Under appropriate convexity assumptions on f and the g_i, this $\tilde{P}$-algorithm has primal-dual properties which lead to valuable lower bounds and to estimates of the Kuhn-Tucker multipliers. The following is assumed to hold in this section only.

Assumption 4. For $i = 1, ..., m$, each function g_i is concave over F^o and f is convex over F^o.
An immediate consequence of this assumption is that, in certain instances, the optimal objective value in DF is a lower bound on min P. Define

$$\tilde{F}(x^*, \delta) = \{x \mid a_i + b_i^t(x-x^*) \geq 0,\ i = 1, ..., m\ ,$$

$$N(x - x^*) \leq \delta\}\ . \tag{3.1}$$

Theorem 4. Let assumption 4 hold and assume that either
(a) The set of points x satisfying $N(x - x^*) \leq \delta$ contains a point which minimizes $P(x, r)$ over F, or
(b) $N(s^*) < \delta$, where s^* solves D$F(x^*)$
Then

$$\min \{\tilde{P}(x; x^*, r) | x \in \tilde{F}(x^*, \delta)\} \leq \min \{P(x, r) | x \in F\}\ . \tag{3.2}$$

Proof. By assumption 4, for any points x, x^* in F^o

$$f(x) \geq a_0 + b_0^t(x-x^*) \tag{3.3}$$

$$g_i(x) \leq a_i + b_i^t(x-x^*),\quad i = 1, ..., m\ . \tag{3.4}$$

By (3.3) and (3.4), the "outer linearization" of F contains F, i.e.,

$$\mathrm{L}F(x^*) \supseteq F \tag{3.5}$$

so

$$\mathrm{L}F^o(x^*) \supseteq F^o \tag{3.6}$$

and $\tilde{P}$ is defined over F^o. Since $B(z)$ is monotone decreasing for $z > 0$, (3.4) implies

$$B(a_i + b_i^t(x-x^*)) \leq B(g_i(x)),\quad i = 1, ..., m \tag{3.7}$$

for any $x \in F^{\circ}$. Hence, using (3.3) and (3.7)

$$\widetilde{P}(x; x^*, r) \leq P(x, r) \qquad \text{for all } x \in F\,. \tag{3.8}$$

Under hypothesis (a), let $x^{\circ} \in \widetilde{F}(x^*, \delta)$ minimize $P(x, r)$. By (3 8)

$$\min\{\widetilde{P}(x; x^*, r) | x \in \widetilde{F}(x^*, \delta)\} \leq \widetilde{P}(x^{\circ}; x^*, r) \leq P(x^{\circ}, r)$$

and the theorem is proved. Under hypothesis (b)

$$\min\{\widetilde{P}(x; x^*, r) | x \in \widetilde{F}(x^*, \delta)\} = \min\{\widetilde{P}(x; x^*, r) | x \in \mathrm{L}F(x^*)\}.$$

By (3.8)

$$\min\{\widetilde{P}(x; x^*, r) | x \in F\} \leq \min\{P(x, r) | x \in F\}\,. \tag{3.9}$$

Since $\mathrm{L}F(x^*) \supseteq F$

$$\min\{\widetilde{P}(x; x^*, r) | x \in \mathrm{L}F(x^*)\} \leq \min\{\widetilde{P}(x; x^*, r) | x \in F\}\,. \tag{3.10}$$

Relations (3.9)–(3.10) prove the theorem.

By theorem 4, min $\widetilde{P}$, in conjunction with the current best feasible point, may be used to terminate computations when the difference between the two values is less than some epsilon. As we will show shortly, if $\widetilde{P}$ is strictly convex there is a subsequence of optimal directions s^* which converges to zero, and the corresponding subsequence of points x^* approaches a P-minimum. It is easily seen that min $\widetilde{P}$ then converges to min P, so the two bounds approach each other.

In addition to this lower bound on min P, each $\widetilde{P}$-minimization for which the normalization constraint is not binding provides an estimate of the Kuhn-Tucker multipliers for NLP, and a lower bound on min f. Both arise from a feasible point for the Wolfe dual of NLP.

Theorem 5. Let s^* solve $\mathrm{D}F(x^*)$ and assume that $N(s^*) < \delta$. Define

$$u_i(x^*) = -rB'(a_i + b_i^t s^*), \quad i = 1, \ldots, m$$

and

$$u(x^*) = (u_1(x^*), \ldots, u_m(x^*))\,.$$

Then $(x^*, u(x^*))$ is feasible for the Wolfe dual of NLP and

$$f(x^*) - u(x^*)g(x^*) \le \min\{f(x) | x \in F\}. \tag{3.11}$$

Proof. The Wolfe dual of NLP is [12]

$$\text{maximize } L(x, u)$$

subject to

$$\nabla_x L(x, u) = 0 \tag{3.12}$$

and

$$u \ge 0 \tag{3.13}$$

where

$$L(x, u) = f(x) - ug(x). \tag{3.14}$$

Since s^* is an unconstrained solution

$$\nabla \widetilde{P}(s^*; x^*, r) = b_0 + r \sum_{i=1}^{m} B'(a_i + b_i^t s^*) b_i = 0$$

or

$$b_0 = \sum_{i=1}^{m} u_i(x^*) b_i. \tag{3.15}$$

Since $B(z)$ is monotone decreasing for $z > 0$

$$B'(a_i + b_i^t s^*) < 0$$

so

$$u_i(x^*) > 0, \quad i = 1, ..., m. \tag{3.16}$$

By (3.15) and (3.16), $(x^*, u(x^*))$ satisfies (3.12) and (3.13), and so is feasible for the Wolfe dual. Further, under the convexity assumptions 4,

$$L(x, u) \le \min\{f(x) | x \in F\}$$

for any dual feasible point (x, u). Evaluating L at $(x^*, u(x^*))$ yields (3.11).

Even when the convexity assumptions are dropped, $u(x^*)$ is a valid estimate of the multipliers, provided only that the optimal solution of DF is unconstrained, so that $\nabla\widetilde{P} = 0$ there. As $r \to 0$, $u_i(x^*)$ will tend to zero for g_i which remain positive, so complementary slackness will hold.

4. Conditioning of DF and strict convexity of $\widetilde{P}$

The second partial derivatives of $\widetilde{P}$ are

$$\frac{\partial^2 \widetilde{P}}{\partial x_i \partial x_j} = r \sum_{k=1}^{m} B''(a_k + b_k^t s)\, b_{ki} b_{kj} \; .$$

Hence the Hessian of P may be written as a linear combination of dyadic terms

$$\nabla^2 \widetilde{P} = r \sum_{k=1}^{m} B''(a_k + b_k^t s)\, b_k b_k^t \; . \tag{4.1}$$

Alternatively, defining

$$D = \text{diag}\,(B''(a_i + b_i^t s))$$

and the Jacobian of the constraints

$$J = \begin{bmatrix} b_1^t \\ \vdots \\ b_m^t \end{bmatrix}$$

we have the expression

$$\nabla^2 \widetilde{P} = r J' DJ \; .$$

Let $B(z) = 1/z$. Then, if s^* solves D$F(x^*)$

$$rB''(a_k + b_k^t s^*) = 2r/(a_k + b_k^t s^*)^3 = 2r^{-1/2}\, u_k(x^*)^{3/2}$$

so

$$\nabla^2 \widetilde{P}(s^*; x^*, r) = 2r^{-1/2} \sum_{k=1}^{m} u_k(x^*)^{3/2}\, b_k b_k^t\,.$$

As $r \to 0$, x^* approaches an optimal solution to NLP and $u_k(x^*)$ approaches the k^{th} Lagrange multiplier. This multiplier tends to zero for inactive constraints g_i. Hence $\nabla^2 \widetilde{P}$, evaluated at optima of DF, approaches

$$M = 2r^{-1/2} \sum_{k \in I} u_k^{3/2} (b_k b_k^t) \tag{4.2}$$

where I is the set of indices of active constraints and u_k is the k^{th} Lagrange multiplier. Similar conclusions are reached for any twice differentiable function B satisfying assumption 3.

If I contains $r < n$ indices, M is positive semidefinite of rank r, hence singular. If Newtons method is used to solve DF, inversion of $\nabla^2 \widetilde{P}$ will become increasingly difficult. A possible remedy for this (which has not yet been tried computationally) is to choose N as the L_2 norm, and to replace the normalization constraint by

$$N^2 = s^t s \leq \delta\,. \tag{4.3}$$

If this constraint is incorporated into the objective by a Lagrange multiplier λ, the Hessian of the augmented objective, L, is

$$\nabla^2 L = \nabla^2 \widetilde{P} + 2\lambda I.$$

As $r \to 0$

$$\nabla^2 L \to 2r^{-1/2} M + 2\lambda I = 2r^{-1/2} (M + \lambda r^{1/2} I)\,.$$

If $\lambda r^{1/2}$ approaches a finite positive value as $r \to 0$ then $\nabla^2 L$ approaches a positive definite matrix with finite condition number (ratio of largest to smallest eigenvalue). This is in contrast to P, since Powell has shown [2] that the condition number of $\nabla^2 P$ approaches infinity if there are less than n binding constraints at the optimum. The above requirement on $\lambda r^{1/2}$ means that the region defined in (4.3) cannot be too large as

$r \to 0$. Since this region should shrink as a solution point is approached, this does not appear to be a serious limitation.

Strict convexity of $\tilde{P}$ is characterized in the following theorem.

Theorem 6. $\tilde{P}$ is strictly convex if and only if $m \geq n$ and the set $\{b_1, \ldots, b_m\}$ contains n independent vectors.

Proof. Let $s_1 \neq s_2$, $0 < \alpha < 1$, and $\bar{\alpha} = 1 - \alpha$. Then

$$\tilde{P}(\alpha s_1 + \bar{\alpha} s_2) = \alpha(a_0 + b_0^t s_1) + \bar{\alpha}(a_0 + b_0^t s_2)$$

$$+ r \sum_{i=1}^{m} B[\alpha(a_i + b_i^t s_1) + \bar{\alpha}(a_i + b_i^t s_2)] \ .$$

By strict convexity of B

$$B[\alpha(a_i + b_i^t s_1) + \bar{\alpha}(a_i + b_i^t s_2)] < \alpha\, B(a_i + b_i^t s_1)$$

$$+ \bar{\alpha}\, B(a_i + b_i^t s_2)$$

if and only if

$$b_i^t s_1 \neq b_i^t s_2 \ .$$

Hence

$$\tilde{P}(\alpha s_1 + \bar{\alpha} s_2) < \alpha\, \tilde{P}(s_1) + \bar{\alpha}\tilde{P}(s_2) \tag{4.4}$$

if and only if

$$b_i^t s_1 \neq b_i^t s_2 \qquad \text{for some } i, \quad i = 1, \ldots, m$$

i.e. if and only if s_1 and s_2 do not satisfy

$$b_i^t s_1 = b_i^t s_2 \ , \quad i = 1, \ldots, m \ .$$

Since s_1 and s_2 are arbitrary, except for $s_1 \neq s_2$, (4.4) holds if and only if there is no s except $s = 0$ such that

$$b_i^t s = 0 \ , \quad i = 1, \ldots, m \ .$$

But this is true if and only if the set $\{b_1, ..., b_m\}$ contains n independent vectors.

Strict convexity of $\widetilde{P}$ is important if a Newton method is to be applied in solving $D\widetilde{F}$, since then $\nabla^2 \widetilde{P}$ is positive definite, hence invertible. By theorem 6, if $\widetilde{P}$ is not initially strictly convex, it can be made so by adding upper or lower bound constraints on each x_i to NLP, where the bounds are chosen large enough so as not to restrict the optimal solution.

5. Convergence

We define a solution to the P-problem as any point x where

$$\nabla P(x, r) = 0 .$$

To prove convergence to such a stationary point, we use the following theorem of Zangwill [9, p. 281].

Convergence theorem. Suppose that the $\widetilde{P}$-algorithm of section 2 satisfies the following conditions

1. If the algorithm terminates, it terminates at a solution
2. If there exists a convergent subsequence

$$x_k \to x^*, \qquad k \in K$$

where x^* is not a solution, and if s_k solves $DF(x_k)$, then there is a $K_1 \subset K$ such that

(a) $s_k \to s^*$, $k \in K_1$
(b) $\nabla P^t(x^*, r)s^* < 0$
(c) A $\delta > 0$ exists such that, for any α satisfying $0 \leq \alpha \leq \delta$

$$x_k + \alpha s_k \in F, \qquad k \in K_1 .$$

Then the algorithm either terminates at a solution or the limit of any convergent subsequence is a solution.

Definition. A point to set map M: $x \to D(x)$ is closed (i.e. upper semicontinuous) at a point x^* if

$$x_k \to x^* , \quad k \in K$$

$$s_k \in D(x_k)$$

and

$$s_k \to s^* , \quad k \in K$$

imply

$$s^* \in D(x^*) .$$

Let

$$D(x) = \{s \mid s \text{ solves } \mathrm{D}F(x)\} \tag{5.1}$$

and let M be the map $M: x \to D(x)$. The following theorem is central to the convergence proof.

Theorem 7. Under assumption 1, the map M defined above is closed at any point $x \in F^{\circ}$.

Proof. Let $x^* \in F^{\circ}$ and choose a sequence of points $\{x_k\}_K$, all in F°, such that

$$x_k \to x^* , \quad k \in K .$$

Let

$$s_k \to s^* , \quad k \in K .$$

where

$$s_k \in D(x_k) \tag{5.2}$$

and assume

$$s^* \notin D(x^*) . \tag{5.3}$$

By definition

$$g_i(x_k) + \nabla g_i^t(x_k) s_k > 0 , \ i = 1, \ldots, m \tag{5.4}$$

$$N(s_k) \leq \delta . \tag{5.5}$$

Since the left hand sides of (5.4) and (5.5) are continuous functions of (x, s), (x^*, s^*) satisfies (5.4) – (5.5), i.e. s^* is feasible for $\mathrm{D}F(x^*)$. By (5.3), there is an s also feasible for $\mathrm{D}F(x^*)$ with lower objective value;

$$g_i(x^*) + \nabla g_i^t(x^*)\hat{s} > 0, \quad i = 1, ..., m \tag{5.6}$$

$$N(\hat{s}) \leq \delta \tag{5.7}$$

and

$$\widetilde{P}(\hat{s}; x^*, r) < \widetilde{P}(s^*; x^*, r) .$$

Define

$$\epsilon = \widetilde{P}(\hat{s}^*; x^*, r) - \widetilde{P}(s; x^*, r) > 0 . \tag{5.8}$$

By continuity of $\widetilde{P}$ in (s, x)

$$\lim_{k \in K} \widetilde{P}(s_k; x_k, r) = \widetilde{P}(\hat{s}^*; x^*, r) .$$

Hence for sufficiently large k

$$|\widetilde{P}(s_k; x_k, r) - \widetilde{P}(s^*; x^*, r)| < \epsilon/2 . \tag{5.9}$$

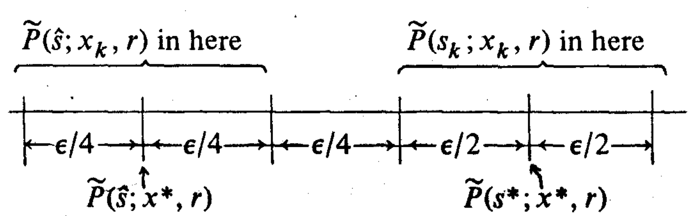
Using the continuity of ∇g_i, (5.6) implies that, for k sufficiently large

$$g_i(x_k) + \nabla g_i^t(x_k)\hat{s} > 0 \tag{5.10}$$

so $\widetilde{P}(\hat{s}; x, r)$ is defined and continuous in x for all x satisfying (5.10). Thus, for sufficiently large k

$$|\widetilde{P}(\hat{s}; x_k, r) - \widetilde{P}(\hat{s}; x^*, r)| < \epsilon/4 . \tag{5.11}$$

Inequalities (5.8), (5.9), and (5.11) may be represented geometrically as follows



Clearly, for k sufficiently large

$$\widetilde{P}(\hat{s}; x_k, r) < \widetilde{P}(s_k; x_k, r) . \tag{5.12}$$

By (5.7) and (5.10), $\hat{s}$ is feasible for D$F(x_k)$ for large k, so (5.12) contradicts the optimality of s_k in D$F(x_k)$. Hence our assumption that $s^* \notin D(x^*)$ must have been false, and the theorem is proved.

We now state and prove the main convergence theorem.

Theorem 8. Under assumptions (1)–(3), the $\widetilde{P}$-algorithm of section 2 either terminates at a point x such that $\nabla P(x, r) = 0$, or the limit of any convergent subsequence is such a point.

Proof. The conditions of Zangwill's convergence theorem will be verified. We assume for purposes of this proof that the termination criterion used is $\nabla P = 0$. Since a downhill direction is generated whenever $\nabla P \neq 0$, condition 1 holds. Since all s_k satisfy the normalization constraint, and hence are in a compact set, condition 2a holds. Consider condition 2b. All points x_k of the subsequence indexed by K_1 are in F^o. The limit point x^* must also be in F^o, because

$$P(x_{k+1}, r) < P(x_k, r)\,,\quad k \in K_1 \tag{5.13}$$

and x_∞ lying on the boundary of F would imply

$$\lim_{k\in K_1} \; P(x_k, r) = +\infty$$

contradicting (5.13). Then, by theorem (7),

$$s^* \in D(x^*)\,.$$

Since, by assumption

$$\nabla P(x^*, r) \neq 0$$

theorem 3 states that condition 2b holds. Turning finally to 2c, since all points x_k, $k \in K_1$ and x^* are in F^o, there exist $\delta_k > 0$ such that, for all $k \in K_1$

$$x_k + \alpha s_k \in F, \quad \text{for all } 0 \leq \alpha \leq \delta_k\,. \tag{5.14}$$

It is convenient to choose δ_k as the Euclidean distance from x_k to the nearest boundary point. If

$$\inf_{k \in K_1} \{\delta_k\} = 0$$

then there is a subsequence of $\{\delta_k\}_{K_1}$ defined by an index set $K_2 \subseteq K_1$ such that

$$\lim_{k \in K_2} \delta_k = 0 .$$

This implies that the subsequence $\{x_k\}_{K_2}$ converges to a point on the boundary of F. But $\{x_k\}_{K_2}$ converges to $x^* \in F^o$, so this is a contradiction and

$$\delta^* = \inf_{k \in K_1} \{\delta_k\} > 0 .$$

Hence (5.14) holds with δ_k replaced by δ^*, condition 2c is satisfied and the theorem is proved.

Under additional hypotheses, more may be said about the limiting behavior of the $\widetilde{P}$-algorithm.

Theorem 9. Let $\{x_k\}_K$ be a convergent subsequence of points generated by the $\widetilde{P}$-algorithm with limit x^* and let s_k solve $DF(x_k)$. If $\widetilde{P}(s; x^*, r)$ is strictly convex, then

$$\lim_{k \in K} s_k = 0 .$$

Proof. By theorem 8

$$\nabla P(x^*, r) = 0 . \tag{5.15}$$

By theorem 6, $\widetilde{P}(s; x^*, r)$ is strictly convex if and only if $m \geq n$ and the set $\{\nabla g_1(x^*), \ldots, \nabla g_m(x^*)\}$ contains n independent vectors. Then, by (4.1), $\nabla^2 \widetilde{P}(s; x^*, r)$ is positive definite for all $s \in LS^o(x^*)$. But $\nabla^2 \widetilde{P}(s; x^*, r)$ is the Jacobian of the system

$$\nabla \widetilde{P}(s; x^*, r) = 0 . \tag{5.16}$$

Since $\widetilde{P}$ is strictly convex and (5.15) holds, (5.16) has the unique solution $s = 0$. Since the Jacobian of (5.16) is nonsingular, the implicit function theorem states that (5.16) has a solution $s(x)$ for all x in some neighborhood of x^*, and $s(x)$ is a continuous function of x. But continuity of $s(x)$ implies

$$\lim_{k \in K} s(x_k) = s(x^*) = 0 .$$

6. Properties with linear constraints

Many nonlinear programs contain some linear constraints. There is substantial evidence [8, 13], that these are best handled by including only the nonlinear constraints in the barrier term, and minimizing the barrier function subject to the linear constraints. Such an option is easily incorporated into the $\widetilde{P}$-algorithm. Let

$$F_l = \{x \mid c_i^t x \le e_i,\ i = 1, ..., r,\ c_i^t x = e_i,\ i = r+1, ..., s\} \quad (6.1)$$

and

$$F_n = \{x \mid g_i(x) \ge 0,\ i = 1, ..., m\} . \quad (6.2)$$

The set F_l is determined by some of the linear constraints of the problem NLP, whose feasible region is $F_n \cap F_l$. The modified P-problem is

$$\text{minimize } P(x, r) = f(x) + r \sum_{i=1}^{m} B(g_i(x))$$

subject to

$$x \in F_n \cap F_l .$$

Any solution to this problem will be in F_n^o, so P must be minimized subject to $x \in F_l$. Minimization of $\widetilde{P}$ must also incorporate this condition, so D$F(x^*)$ is most easily written in terms of x, rather than $s = x - x^*$. The modified D$F(x^*)$ is

$$\text{minimize } \widetilde{P}(x; x^*, r)$$

subject to

$$x \in \widetilde{F}(x^*, \delta) \cap F_l$$

where $\widetilde{F}(x^*, \delta)$ is defined in eq. (3.1).

If x^o solves the above and $s^* = x^o - x^*$, then s^* is obviously a feasible direction for P at x^*. If the L_∞ norm is used in DF, then $\widetilde{F} \cap F_l$ is determined by linear constraints. Thus any method which can solve linearly constrained problems (e.g. that of Goldfarb [14] or McCormick [15]) may be used to solve DF.

All results derived earlier hold for this modified version of DF if they are rephrased appropriately. For example, theorem 2 now states that x^* solves D$F(x^*)$ if and only if x^* is a Kuhn-Tucker point for the modified P-problem. Theorem 3 also applies, with condition (a) modified to state that x^* is not a Kuhn-Tucker point for the P-problem, and the added assumption that $x^* \in F_l$. The proofs are much the same. The only significant change is to modify the optimality conditions for DF and the P-problem to accomodate the linear constraints. Similar comments apply to theorems 4 and 5. Turning to the convergence theorem, we define a solution to be a Kuhn-Tucker point for the P-problem. Theorem 7 is true with F^o replaced by $F_n^o \cap F_l$, and theorem 8 holds with the new definition of solution point. Again the proofs require only minor modification, and will not be redone here.

7. Extension to quasi-barrier and exterior penalty functions
Quasi-barrier functions

Allran and Johnsen [16] propose solving NLP by successive unconstrained minimization of the function

$$P(x, n) = f(x) + \sum_{i=1}^{m} \exp(-T_{in}\, g_i(x))$$

where

$$0 < T_{in} < T_{i,n+1}\ , \quad i = 1, \ldots, m$$

and

$$\lim_{n \to \infty} T_{in} = +\infty\ .$$

They prove that, for sufficiently large n, $P(x, n)$ has an unconstrained minimum in F^o. The hypotheses under which this is true are very similar to those for barrier functions. Convergence of $\{\min P(x, n)\}$ to $\min f$ and of the sequence of minimizing points to an optimal point is also proved.

The function

$$Q(Tz) = e^{-Tz}, \;\; T > 0$$

is not a barrier function for any finite T, but approaches one as $T \to +\infty$. The $\widetilde{P}$ function corresponding to $P(x; n)$, with base point x^*, is

$$\widetilde{P}(s; x^*, n) = a_0 + b_0^t s + \sum_{i=1}^{m} \exp\,[-\,T_{in}\,(a_i + b_i^t s)]$$

where

$$s = x - x^*\,.$$

As with barrier functions, $\widetilde{P}$ may not have an unconstrained minimum even if P does, so a normalization condition must be included in D$F(x^*)$. In contrast to previous sections, P is defined for all $s \in E^n$, so the problem D$F(x^*)$ is

$$\text{minimize } \widetilde{P}(s; x^*, n)$$

subject to

$$N(s) \leq \delta\,.$$

$Q(Tz)$ satisfies the conditions of assumption 3 for all real z. Hence, it is easily verified that all theorems and results of previous sections apply.

Theorem 7 now states that the map M is closed at any point $x \in E^n$. The proofs, especially those in theorems 7 and 8 are simplified, since $P(x, n)$ is defined over all of E^n. Thus the $\widetilde{P}$-algorithm is a valid approach for minimizing quasi-barrier functions.

Exterior penalty functions

Here we focus attention on the penalty function

$$P(z) = (\min(0, z))^2\,. \tag{7.1}$$

This is used to solve NLP by unconstrained minimization of

$$G(x, k) = f(x) + k \sum_{i=1}^{m} P(g_i(x)) \tag{7.2}$$

where $k > 0$ and $k \to +\infty$. Properties of the method are given in [9]. The associated problem $DF(x^*)$ is

$$\text{minimize } a_0 + b_0^t s + k \sum_{i=1}^{m} P(a_i + b_i^t s) = \widetilde{G}(s; x^*, k) \qquad (7.3)$$

subject to

$$N(s) \leq \delta \,.$$

Graphs of $P(z)$ and $P'(z)$ are shown in fig. 1.

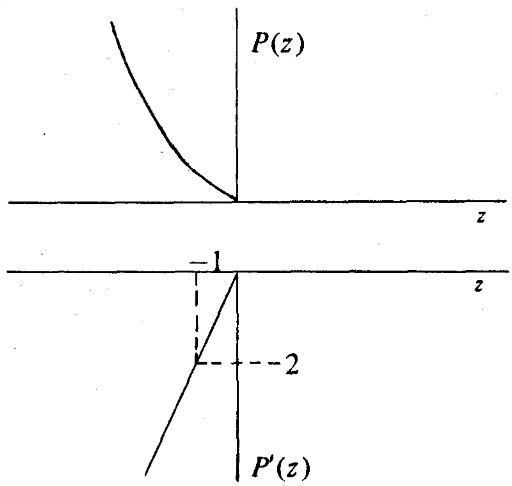


Fig. 1. $P(z)$ and $P'(z)$.

Since P is convex, theorems 1 and 2 still hold. The monotonicity conditions of assumption 3, however, apply only for $z < 0$. Although P is differentiable for all z, P and P' are identically zero for $z \geq 0$. Despite this, theorem 3 is still true, as we now show.

Theorem 10. For $P(z)$ and $G(x, k)$ in (7.1)–(7.2) assume
(a) $\nabla G(x^*, k) \neq 0$
(b) s^* solves $DF(x^*)$.
Then

$$\nabla G^t(x^*, k)\, s^* < 0 \,.$$

Proof. Lemmas 1 and 2 of section 2 and the saddle point conditions may be applied here as in the proof of theorem 3 to yield

$$\nabla \widetilde{G}(s^*; x^*, k) + \lambda^* y = 0$$

where

$$y \in \partial N(s^*) .$$

Writing out the expression for $\nabla \widetilde{G}$ yields

$$b_0 + k \sum_{i=1}^{m} P'(a_i + b_i^t s^*) b_i + \lambda^* y = 0 . \tag{7.3a}$$

By (7.2)

$$\nabla G^t(x^*, k) s^* = b_0^t s^* + k \sum_{i=1}^{m} P'(a_i) b_i^t s^* . \tag{7.4}$$

Taking the scalar product of (7.3a) with s^* yields

$$b_0^t s^* = -k \sum_{i=1}^{m} P'(a_i + b_i^t s^*) b_i^t s^* - \lambda^* y^t s^* . \tag{7.5}$$

Substituting (7.5) into (7.4)

$$\nabla G^t(x^*, k) s^* = k \sum_{i=1}^{m} (P'(a_i) - P'(a_i + b_i^t s^*)) b_i^t s^* - \lambda^* y^t s^* . \tag{7.6}$$

As in theorem 3, we may conclude that

$$y^t s^* \geq 0$$

so the last term on the right of (7.6) is non-positive. Consider now the term

$$t_i = (P'(a_i) - P'(a_i + b_i^t s^*)) b_i^t s^* . \tag{7.7}$$

Using the monotonicity of P', it is easily seen that

$$a_i < 0 \Rightarrow t_i < 0 \quad \text{if} \quad b_i^t s^* \neq 0 \tag{7.8}$$

$$a_i \geq 0 \Rightarrow \begin{cases} t_i = 0 & \text{if } a_i + b_i^t s^* \geq 0 \\ t_i < 0 & \text{if } a_i + b_i^t s^* < 0 . \end{cases} \tag{7.9}$$

Hence each t_i is non-positive. We will show that at least one t_i is negative. By (7.8)–(7.9) there are only two cases under which all t_i are zero:

(1) $b_i^t s^* = 0, \ i = 1, ..., m$

(2) for all i such that $b_i^t s^* \neq 0$, $a_i \geq 0$ and $a_i + b_i^t s^* \geq 0$.

By showing $\nabla G^t(x^*, k)s^* < 0$ for these cases, we prove the theorem. Assume that case 1 holds. Then

$$\begin{aligned} \widetilde{G}(s^*; x^*, k) &= a_0 + k \sum_{i=1}^{m} P(a_i) + b_0^t s^* \\ &= \widetilde{G}(0; x^*, k) + b_0^t s^* \end{aligned} \tag{7.10}$$

while, by (7.4)

$$\nabla G^t(x^*, k) s^* = b_0^t s^* . \tag{7.11}$$

Under case 2

$$I = \{i \mid b_i^t s^* \neq 0\} \neq \emptyset$$

and

$$\begin{aligned} \widetilde{G}(s^*; x^*, k) &= a_0 + b_0^t s^* + k \sum_{i \notin I} P(a_i) + k \sum_{i \in I} P(a_i + b_i^t s^*) \\ &= a_0 + k \sum_{i \notin I} P(a_i) + b_0^t s^* . \end{aligned}$$

Since $a_i \geq 0, i \in I, P(a_i) = 0, i \in I$, so

$$\begin{aligned} \widetilde{G}(s^*; x^*, k) &= a_0 + k \sum_{i=1}^{m} P(a_i) + b_0^t s^* \\ &= \widetilde{G}(0; x^*, k) + b_0^t s^* \end{aligned}$$

as in case 1, eq. (7.10). Relation (7.11) also holds for case 2, so we focus on (7.10)–(7.11).

If $b_0^t s^* > 0$, then $\widetilde{G}$ can be reduced by setting $s^* = 0$, which contradicts the optimality of s^*. If $b_0^t s^* = 0$, then $s^* = 0$ is optimal for $DF(x^*)$. By theorem 2, this contradicts hypothesis (a) of the theorem. Hence $b_0^t s^* <$ 0 and, by (7.11), the theorem is proved.

Application of the $\widetilde{G}$-algorithm to exterior penalty functions is perhaps even more attractive computationally than applying it to barrier functions. This is because $\widetilde{G}$ in (7.3) is piecewise quadratic, the pieces being polyhedral regions where various subsets of the linearized constraints are negative. Hence, if N is the L_∞ norm, DF is easily transformed into a quadratic program, which can be solved in a finite number of pivot steps. Other efficient schemes also exist. Theoretical and computational work on this exterior penalty case is now in progress, and will be the subject of a future paper.

8. Computational results

To evaluate the efficiency of this $\widetilde{P}$ algorithm, 7 test problems were solved. These had from 2 variables and 2 constraints to 15 variables and 20 constraints, and are specified in appendix 1. All have linear or quadratic constraints, and quadratic or cubic objective functions. In problems 1–4, DF was solved by separable programming with column generation. The details of this approach are as follows. New variables t_i are introduced, and DF is re-expressed as

$$\text{minimize } b_0^t s + r \sum_{i=1}^{m} B(t_i) \tag{8.1}$$

subject to

$$0 \leq t_i \leq a_i + b_i^t s\,, \quad i = 1, ..., m \tag{8.2}$$

and

$$-\delta \leq s_i \leq \delta\,, \quad i = 1, ..., n\,. \tag{8.3}$$

Where N has been chosen as the L_∞ norm. Since B is decreasing, t_i will equal $a_i + b_i^t s$ in any optimal solution of (8.1)–(8.3). Suppose now that, for each i, a set of grid points $\{t_{ij}\}$ is chosen and $B(t_i)$ is replaced by its piecewise linearization over this grid:

$$B(t_i) = \sum_j \lambda_{ij} B(t_{ij})$$

where

$$t_i = \sum_j \lambda_{ij} t_{ij}$$

$$\sum_j \lambda_{ij} = 1, \quad \lambda_{ij} \geq 0 .$$

DF in (8.1)–(8.3) is then transformed into an approximating linear program in the variables λ_{ij}:

$$\text{minimize } b_0^t s + r \sum_{i,j} \lambda_{ij} B(t_{ij}) \tag{8.4}$$

subject to

$$-b_i^t s + \sum_j \lambda_{ij} t_{ij} + r_i = a_i, \quad i = 1, \ldots, m \tag{8.5}$$

$$\sum_j \lambda_{ij} \qquad = 1, \quad i = 1, \ldots, m \tag{8.6}$$

$$-\delta \leq s_i \leq \delta \qquad , \quad i = 1, \ldots, m \tag{8.7}$$

$$\lambda_{ij} \geq 0 , \quad \text{all } i, j . \tag{8.8}$$

An initial basic feasible solution is

$$\lambda_{i1} = 1, \quad t_{i1} = r_i = 0.5\, a_i, \quad i = 1, \ldots, m$$

whose associated basis matrix is triangular. The bounds (8.7) can be dealt with by upper bounding methods.

Instead of choosing the grid points in advance, they can be generated via subproblems. Assume a feasible basis for (8.4)–(8.8) is available, and let u_i and v_i be the simplex multipliers of this basis, with u_i associated with (8.5) and v_i with (8.6). The relative cost factor for λ_{ij} is

$$\bar{c}_{ij} = r\, B(t_{ij}) - u_i t_{ij} - v_i . \tag{8.9}$$

The standard simplex criterion is to search for that grid point yielding minimal $\bar{c}_{ij}$. This leads to the subproblem

$$\text{minimize } rB(t_i) - u_i t_i \tag{8.10}$$

subject to

$$t_i \geq 0 . \tag{8.11}$$

For this to have an optimal solution with $t_i > 0$ it is necessary and sufficient that

$$rB'(t_i) = u_i . \tag{8.12}$$

Since B is monotone decreasing, (8.12) has a solution if and only if $u_i < 0$. The condition $u_i \leq 0$ can always be guaranteed, since if $u_i \geq 0$, a slack variable can enter the basis. Assuming $u_i < 0$, (8.12) has the solution

$$t_i = (-r/u_i)^{1/2} \quad \text{if} \quad B(t) = 1/t$$

$$t_i = -r/u_i \quad \text{if} \quad B(t) = -\ln(t)$$

The grid point with the most negative relative cost factor is used to form a column, which is brought into the basis. Solutions for $u_i = 0$ are also easily derived. Dantzig [17] proves that this algorithm converges in the limit.

In test problems 5–6, DF was solved by Goldfarb's modification of Davidon's method to account for linear constraints [14]. This was adapted to the special case of upper and lower bounds, yielding significant simplifications. The linear search required by Goldfarb's algorithm was accomplished by a regula falsi procedure. The linear search required after a direction of travel is found by DF was done by cubic interpolation, similar to the procedure outlined in [18]. The termination criteria for this linear search were to stop when

$$\cos\theta = \frac{|s^t g|}{\|s\| \cdot \|g\|} < 10^{-3}$$

or when 3 cubic interpolations have been made. For purposes of comparison, all test problems except that in table 2 were also solved using the Davidon–Fletcher–Powell algorithm [18] to minimize $P(x, r)$. The version used was restarted every $n + 1$ cycles, as suggested in [15] and [19], by resetting the H matrix to the identity, and used the same one-dimensional search as the $\widetilde{P}$-algorithm. All computations were done on the Univac 1108 computer in single precision arithmetic (8 decimal digit word length), and all algorithms were coded in FORTRAN 5.

The number of cycles given in tables 1–12 for both $\widetilde{P}$ and Davidon algorithms are those required for P to become less than or equal to the numbers in the min P column. These numbers are equal to the final P values obtained to 4 or 5 significant figures. Since all computations carried only 8 decimal digits, and since the directions produced by DF were probably correct only to 2 or 3 digits, it was felt that these figures best represented the true performance of both algorithms. In general, any iterations beyond those listed made little or no progress in reducing P.

Tables 4–6 illustrate 2 different strategies for choosing δ in $N(s) \leq \delta$. Tables 4–5 use a constant value for δ. In table 6, if α_i is the optimal step size value at iteration i, then δ was replaced by $\delta/1.5$ if $\alpha_i < 0.8$, by 1.5δ if $\alpha_i > 1.2$, and was unchanged otherwise. The rationale here is that α_i values near unity indicate that $\widetilde{P}$ approximates P well over the set defined by $N(x - x_i) \leq \delta$, since the actual step size to the minimum of P along s_i is nearly that predicted by $\widetilde{P}$. If the α_i are less than one, the region of linearization is too large, and conversely if $\alpha_i > 1$. This simple strategy produced the best results in problems 3, 4 and 5 (all cycles in problem 4 had $N(s) < \delta$), but led to poor results (84 cycles for $r = 1$) in problem 6, where it decreased δ prematurely. There, a constant δ worked much better. It appears that some method for decreasing δ as min P is approached is desirable, and that it should be based on the behavior of the sequence of α_i values. However, a more sophisticated rule is needed.

In problem 5 for $r = 1$, separable programming required approximately 200 pivots to solve each of the first two direction finding problems, and terminated trying to take the logarithm of a negative number in direction finding problem 3. Hence the Davidon algorithm for bounded variables was adopted. This led to much more rapid convergence in DF. In problem 5, computations in DF were terminated when

$$l_i < s_i < u_i \Rightarrow |\partial \widetilde{P}/\partial s_i| < \epsilon$$

$$s_i = l_i \Rightarrow \partial \widetilde{P}/\partial s_i \geq 0$$

$$s_i = u_i \Rightarrow \partial \widetilde{P}/\partial s_i \leq 0$$

where $\epsilon = 10^{-2}$. In problem 6, this led to long computation times in DF (see the column $|G| < 10^{-2}$ in table 12). Hence an additional criterion, which terminated computations when

$$\% \Delta F = \frac{|\widetilde{P}_{i+1} - \widetilde{P}_i|}{|\widetilde{P}_i|} < \eta$$

for 5 consecutive values of i. Table 12 shows results for $\eta = 10^{-2}$ and $\eta = 10^{-4}$. Both result in a much smaller number of DF iterations. $\eta = 10^{-2}$ appears too loose a condition, since min P is higher than in the other two cases.

Table 1
Problem 1, $B(z) = 1/z$

r	Cycles, $\widetilde{P}$	Cycles, Davidon	Avg. DF cycles	min P	$\|\nabla P\|$
1	9	3	7	5.3466	$< 2 \times 10^{-3}$
10^{-1}	4	2	7	2.1475	$< 2 \times 10^{-3}$
10^{-2}	3	2	5.6	1.3388	$< 2 \times 10^{-3}$
10^{-3}	3	2	5.6	1.1045	$< 2 \times 10^{-3}$

Table 2
Problem 1, $B(z) = -\ln(z)$

r	Cycles, $\widetilde{P}$	Avg. DF cycles	min P	$\|\nabla P\|$
1	6	5.2	3.1990	0.76×10^{-2}
10^{-1}	3	6.0	1.5961	0.15×10^{-3}
10^{-2}	2	5.0	1.1042	0.20×10^{-2}
10^{-3}	2	5.0	1.1050	0.23×10^{-2}

Table 3
Problem 2, $B(z) = 1/z$

r	Cycles, $\tilde{P}$	Cycles, Davidon	Avg. DF cycles	min P
1	3	6	39	10.362
10^{-1}	2	4	25	4.1244
10^{-2}	2	3	18	2.2544
10^{-3}	1	3	13	1.6781
10^{-4}	1	2	7	1.4975

Table 4
Problem 3, $B(z) = 1/z$, $\delta = 0.5$

r	Cycles, $\tilde{P}$	Avg. DF cycles	min P	$\|P\|$
1	20	36	– 38.136	4.3
10^{-1}	9	22	– 42.333	0.92
10^{-2}	9	17	– 43.476	1.4
10^{-3}	8	14	– 43.810	2.2
10^{-4}	7	10	– 43.912	3.1
10^{-5}	6	9	– 43.944	5.5
10^{-6}	4	9	– 43.955	8.5

Table 5
Problem 3, $B(z) = 1/z$, $\delta = 0.1$

r	Cycles, $\tilde{P}$	Avg. DF cycles	min P	$\|P\|$
1	11	37	– 38.179	0.49
10^{-1}	2	28	– 42.344	0.26
10^{-2}	7	19	– 43.501	2.2
10^{-3}	4	13	– 43.845	1.9
10^{-4}	6	8	– 43.950	1.8
10^{-5}	6	8	– 43.983	2.4
10^{-6}	4	9	– 43.993	2.7

Table 6
Problem 3, $B(z) = 1/z$, δ variable

r	Cycles, $\tilde{P}$	Cycles, Dav.	Avg. DF cycles	min P	Initial δ	Cycles δ binding	Final δ
1	11	7	35	– 38.179	0.5	11	0.195×10^{-4}
10^{-1}	2	3	30	– 42.344	0.1	2	0.1
10^{-2}	2	2	18	– 43.501	0.1	2	0.1
10^{-3}	3	3	13	– 43.845	0.1	3	0.444×10^{-1}
10^{-4}	3	2	10	– 43.951	0.1	3	0.444×10^{-1}
10^{-5}	3	4	8	– 43.984	0.1	3	0.444×10^{-1}
10^{-6}	3	–	7	– 43.994	0.1	3	0.444×10^{-1}

Table 7
Problem 4, $B(z) = 1/z$

r	Cycles $\tilde{P}$	Cycles, Davidon	Avg. DF cycles	min P	$\lVert P \rVert$
1	2	15	108	– 30505	0.93
10^{-1}	2	10	35	– 30603	0.53×10^{-2}
10^{-2}	2	2	35	– 30633	0.46×10^{-1}
10^{-3}	1	10	21	– 30643	0.66
10^{-4}	1	5	11	– 30646	0.65
10^{-5}	1	5	11	– 30647.3	1.2
10^{-6}	1	4	11	– 30647.6	3.6
10^{-7}	2	2	15	– 30647.7	10.8

Table 8
Problem 4, objective sign reversed

r	Cycles $\tilde{P}$	Cycles, Davidon	Avg. DF cycles	min P	$\lVert P \rVert$
1	2	15	90	23162	0.28
10^{-1}	1	6	53	23090	0.20
10^{-2}	2	6	38	23068	0.03
10^{-3}	1	6	18	23061	2.4
10^{-4}	1	6	9	23058.9	3.0
10^{-5}	1	4	9	23058.2	5.5
10^{-6}	1	5	9	23058.0	3.2
10^{-7}	1	5	9	23057.9	22

Table 9
Problem 5, $B(z) = -\ln(z)$, δ constant

r	Cycles, $\tilde{P}$	Cycles, Davidon	Avg. DF cycles	min P	$\lVert P \rVert$	δ
1	10	11	12	12.424	0.45	0.5
10^{-2}	9	16	16	0.20986	0.92×10^{-2}	0.1
10^{-4}	4	11	10	0.39220×10^{-2}	0.30×10^{-4}	0.1
10^{-6}	2	15	8	0.57642×10^{-4}	0.26×10^{-5}	0.1

Table 10
Problem 5, $B(z) = -\ln(z)$, δ variable

r	Cycles, $\tilde{P}$	Cycles, Dav.	Avg. DF Cycles	min P	$\lVert P \rVert$	Initial	Cycles δ binding	Final δ
1	10	11	13.5	12.422	0.68×10^{-1}	0.5	4	0.29×10^{-2}
10^{-2}	9	16	16	0.20985	0.77×10^{-3}	0.1	9	0.31×10^{-2}
10^{-4}	2	11	16	0.39237×10^{-2}	0.78×10^{-3}	0.1	0	0.1
10^{-6}	2	15	5.5	0.57657×10^{-4}	0.25×10^{-4}	0.1	0	0.1

Table 11
Problem 6, Davidon results

r =	1	0.25	0.0625	0.015625	0.0039	0.97656×10^{-3}
Cycles Davidon	94	22	29	15	16	7
min P	49.515	40.804	35.452	33.373	32.667	32.383

Table 12
Problem 6, $B(z) = -\ln(z)$

r	δ	10	10	20
	DF stop	$\|G\| < 10^{-2}$	$\%\Delta F < 10^{-2}$	$\%\Delta F < 10^{-4}$
1	Cycles, $\tilde{P}$	27	38	25
	Avg. DF cycles	39	9	33
	min P	49.515	49.906	49.519
0.25	Cycles $\tilde{P}$	14	42	21
	Avg. DF cycles	48	6	21
	min P	40.775	41.206	40.777
0.0625	Cycles, $\tilde{P}$	32	18	22
	Avg. DF cycles	65	5	25
	min P	35.462	36.448	35.459
0.015625	Cycles, $\tilde{P}$		17	9
	Avg. DF cycles		5	20
	min P		34.663	33.378
0.0039	Cycles, $\tilde{P}$		15	9
	Avg. DF cycles		5	20
	min P		34.070	32.673
0.97656×10^{-3}	Cycles, $\tilde{P}$		6	6
	Avg. DF cycles		5	5
	min P		33.892	32.457

9. Summary and conclusions

The computational results indicate that, for the problems solved, the search directions produced by DF are as good as, and in some cases significantly better than, those of the Davidon procedure. This is especially true in problem 6, $r = 1$, and problems 4 and 5. However, the computation time required was from the same as to 3 to 4 times greater than that required by the Davidon procedure. This is due to the relatively long times required to solve DF. Clearly a more efficient procedure must be developed if the $\widetilde{P}$ algorithm is to be competitive for minimizing barrier functions. As discussed in section 4, if the L_2 norm is used, then solving DF can be accomplished by unconstrained minimization of a function, L, with positive definite Hessian. In this case, Newton's method should be the best choice. The matrix $\nabla^2 L$ is the identity plus a sum of outer products. As shown by Fiacco and McCormick [4], this structure can be exploited to considerably simplify the inversion process. Use of Newton's method should significantly reduce the time required to solve DF while increasing the accuracy of the solution.

Another possibility yet to be studied is to solve DF only partially. The limiting case of this strategy is to take only one step of Newton's method in each DF (with initial point $s = 0$). If the L_2 norm is used and the constraint $s^t s \leq \delta$ is incorporated with a Lagrange multiplier λ, then search directions s_i are given by

$$s_i = -(\nabla^2 L(0; x_i, r, \lambda))^{-1} \nabla L(0; x_i, r, \lambda)$$

where

$$L(s; x_i, r, \lambda) = \widetilde{P}(s; x_i, r) + \lambda s^t s\,.$$

Using the definition of L and (2.17)

$$\nabla^2 L(0; x_i, r, \lambda) = \nabla^2 \widetilde{P}(0; x_i, r) + 2\lambda I$$

$$\nabla L(0; x_i, r, \lambda) = \nabla \widetilde{P}(0; x_i, r) = \nabla P(x_i, r)$$

so

$$s_i = -(\nabla^2 \widetilde{P}(0; x_i, r) + 2\lambda I)^{-1} \nabla P(x_i, r) \tag{9.1}$$

By (9.1), s_i is given by a formula like Newton's method, but with $\nabla^2 P + 2\lambda I$ replacing $\nabla^2 P$. Since $\nabla^2 \widetilde{P} + 2\lambda I$ is positive definite for all $\lambda > 0$, s_i is a direction of descent for P at x_i. The step size α_i could either be determined by a one-dimensional search or could be controlled by varying λ, as in [23]. Some recent work by Fletcher [25], whose algorithm is similar in its basic philosophy to ours, should be useful in determining how λ should be varied.

The $\widetilde{P}$ approach has some advantages other than requiring fewer cycles. It requires only first derivatives, and the search direction s_i is independent of all past history of the process. This is true of no other efficient first order algorithm. Because of this, the $\widetilde{P}$ algorithm does not require an accurate one dimensional search, as does the Davidon procedure. This fact was not exploited in the work done thus far, but should be of much significance in reducing computation time. Its lack of memory should also make the $\widetilde{P}$ algorithm relatively insensitive to numerical error, at least of the cumulative variety. Hence it may function well with finite difference derivatives, although this has not yet been investigated.

Of course, the features cited above apply also when the $\widetilde{P}$-algorithm is applied to exterior penalty functions. The additional ability to solve DF finitely here suggests that the approach has significant potential. Research, both theoretical and computational, is currently in progress on this extension.

Appendix 1 – Test problems (all constraints ≥ 0)

Problem 1

$$\min \quad f(x) = (x_1 - 2)^2 + (x_2 - 1)^2,$$
$$g_1(x) = -x_1^2 + x_2, \qquad g_2(x) = -x_1 - x_2 + 2,$$
$$x_0 = (-0.5,\ 0.5) = (x_{1,0}\ x_{2,0}).$$

Problem 2

$$\min \quad f(x) = x_1^3 - 6x_1^2 + 11x_1 + x_3,$$
$$g_1(x) = -x_1^2 - x_2^2 - x_3^2, \qquad g_2(x) = x_1^2 + x_2^2 + x_3^2 - 4,$$
$$g_3(x) = -x_3 + 5, \quad g_4(x) = x_1, \quad g_5(x) = x_2, \quad g_6(x) = x_3,$$
$$x_0 = (0.1,\ 0.1,\ 3).$$

Problem 3 Kowalik and Osborne, ref. [20], p. 981

$$\min \quad f(x) = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4,$$

$$g_1(x) = -x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8\,,$$

$$g_2(x) = -x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10\,,$$

$$g_3(x) = -2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5\,,$$

$$g_4(x) = 5 - x_1\,, \qquad g_5(x) = 5 - x_2\,,$$

$$x_0 = (0, 0, 0, 0)\,.$$

Problem 4 Colville, ref. [21],

$a_1 = 5.357$	$a_2 = 0.835$	$a_3 = 37.293$	$a_4 = -40792$
$a_5 = 85.334$	$a_6 = 0.568E{-2}$	$a_7 = 0.626E{-3}$	$a_8 = -220E{-3}$
$a_9 = 80.512$	$a_{10} = 0.00713$	$a_{11} = 0.00299$	$a_{12} = 0.00218$
$a_{13} = 9.300$	$a_{14} = 0.00470$	$a_{15} = 0.00125$	$a_{16} = 0.00190$

$$\min \quad f = a_1 x_3^2 + a_2 x_1 x_5 + a_3 x_1 + a_4\,,$$

$$r_1 = a_5 + a_6 x_2 x_5 + a_7 x_1 x_4 + a_8 x_3 x_5\,,$$

$$r_2 = a_9 + a_{10} x_2 x_5 + a_{11} x_1 x_2 + a_{12} x_3^2\,,$$

$$r_3 = a_{13} + a_{14} x_3 x_5 + a_{15} x_1 x_3 + a_{16} x_3 x_4\,.$$

Constraints (*all* ≥ 0):

$$g_1(x) = r_1(x)\,, \qquad g_2(x) = 92 - r_1(x)\,,$$

$$g_3(x) = r_2(x) - 90\,, \qquad g_4(x) = 110 - r_2(x)\,,$$

$$g_5(x) = r_3(x) - 20\,, \qquad g_6(x) = 25 - r_3(x)\,,$$

$$g_7(x) = x_1 - 78\,, \qquad g_8(x) = 102 - x_1\,,$$

$$g_9(x) = x_2 - 33\,, \qquad g_{10}(x) = 45 - x_2\,,$$

$$g_{11}(x) = x_3 - 27\,, \qquad g_{12}(x) = 45 - x_3\,,$$

$$g_{13}(x) = x_4 - 27\,, \qquad g_{14}(x) = 45 - x_4\,,$$

$$g_{15}(x) = x_5 - 27\,, \qquad g_{16}(x) = 45 - x_5\,,$$

$$x_0 = (78.62, 33.44, 31.07, 44.18, 35.32)\,.$$

Problem 5 Pearson, ref. [22], appendix B

max $f(x) = \frac{1}{2}\,[x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7]$,

$g_1 = 1 - x_3^2 - x_4^2$, $g_2 = 1 - x_9^2$,

$g_3 = 1 - x_5^2 - x_6^2$, $g_4 = 1 - x_1^2 - (x_2 - x_9)^2$,

$g_5 = 1 - (x_1 - x_5)^2 + (x_2 - x_6)^2$, $g_6 = 1 - (x_1 - x_7)^2 + (x_2 - x_8)^2$,

$g_7 = 1 - (x_3 - x_5)^2 - (x_4 - x_6)^2$, $g_8 = 1 - (x_3 - x_7)^2 - (x_4 - x_8)^2$,

$g_9 = 1 - x_7^2 - (x_8 - x_9)^2$, $g_{10} = x_1x_4 - x_2x_3$,

$g_{11} = x_3x_9$, $g_{12} = -x_5x_9$,

$g_{13} = x_5x_8 - x_6x_7$, $g_{14} = x_9$.

Initial point:

$x_0 = (0.433, 0.25, 0.433, 0.75, -0.433, 0.75, -0.433, 0.25, 0.99999)$.

Problem 6 Pearson [22], appendix B (shell dual problem)

min $$-f(x) = -\sum_{j=1}^{10} b_jy_j + \sum_{i=1}^{5}\sum_{j=1}^{5} c_{ij}x_ix_j + 2\sum_{i=1}^{5} d_ix_i^3 .$$

$$g_i(x) = e_i + 2\sum_{j=1}^{5} c_{ji}x_j + 3d_ix_i^2 - \sum_{j=1}^{10} a_{ji}y_j \geq 0, \quad i = 1, \ldots, 5$$

$$x_i \geq 0, \quad i = 1, \ldots, 5$$

$$y_i \geq 0, \quad i = 1, \ldots, 10 .$$

Data for problem 6:

	j \ i	1	2	3	4	5	
e_j	1	−15	−27	−36	−18	−12	
c_{ij}	1	30	−20	−10	32	−10	
	2	−20	39	− 6	−31	32	
	3	−10	− 6	10	− 6	−10	
	4	32	−31	− 6	39	−20	
	5	−10	32	−10	−20	30	
d_j		4	8	10	6	2	b_i
a_{ij}	1	−16	2	0	1	0	−40
	2	0	− 2	0	0.4	2	− 2
	3	− 3.5	0	2	0	0	− 0.25
	4	0	− 2	0	− 4	− 1	− 4
	5	0	− 9	− 2	1	− 2.8	− 4
	6	2	0	− 4	0	0	− 1
	7	− 1	− 1	− 1	− 1	− 1	−40
	8	− 1	− 2	− 3	− 2	− 1	−60
	9	1	2	3	4	5	5
	10	1	1	1	1	1	1

Initial point:

$$x_i = 10^{-4}, \quad i = 1, \ldots, 5,$$

$$y_i = 10^{-4}, \quad i = 1, \ldots, 10, \quad i \neq 7,$$

$$y_7 = 60.$$

References

[1] *Optimization*, Ed. R. Fletcher (Academic Press, 1969) p. 203.

[2] *Ibid.*, p. 205.

[3] D.G. Luenberger, "A combined penalty function and gradient projection method for nonlinear programming," Internal Memorandum 70–2 (Department of Engineering – Economic Systems, Stanford University, June 1970),

[4] A.V. Fiacco and G.P. McCormick, *Nonlinear programming; sequential unconstrained minimization techniques* (John Wiley and Sons, Inc., 1968) Ch. 8,

[5] Fletcher, *op. cit.*, p. 324.

[6] *Ibid.*, pp. 324–325.

[7] *Ibid.*, pp. 209–213.

[8] A.V. Fiacco and G.P. McCormick, *op. cit.*.

[9] W. Zangwill, *Nonlinear programming: a unified approach* (Prentice Hall, 1969).

[10] L. Lasdon, *Optimization theory for large systems* (The Macmillan Company, 1970) Appendix 2.

[11] S. Karlin, *Mathematical models and theory in games programming and economics*, Vol. I Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1959).

[12] P. Wolfe, "A duality theorem for nonlinear programming," *Quarterly of Applied Mathematics* 19, No. 3 (1961) 239–244.

[13] G. Zoutendijk, "Nonlinear programming: a numerical survey," *SIAM Journal on Control* 4 (1966) 194–210.

[14] D. Goldfarb, "Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints," *SIAM Journal on Applied Mathematics* 17, No. 4 (1969) 739–764.

[15] G.P. McCormick, "The variable reduction method for nonlinear programming," *Management Science* 17, No. 3 (1970) 146–160.

[16] R.R. Allran and S.E.J. Johnsen, "An algorithm for solving nonlinear programming problems, subject to nonlinear inequality constraints," *The Computer Journal* 13, No. 2 (1970).

[17] G.B. Dantzig, *Linear programming and extensions* (Princeton University Press, Princeton, New Jersey, 1963) Chs. 22 and 24.

[18] L. Lasdon, *op. cit.*, Ch. 1.

[19] J.D. Pearson, "Variable metric methods of minimization," *British Computer Journal* 12 (1969).

[20] J. Kowalik, M.R. Osborne and D.M. Ryan, "A new method for constrained optimization problems," *Operations Research* 17, No. 6 (1969) 973–983.

[21] A.R. Colville, "Nonlinear programming study results as of November 1968," IBM New York Scientific Center, New York Technical Report No. 320–2949 (1968).

[22] J.D. Pearson, "On variable metric methods of minimization," Technical Paper RAC–TP-302, Research Analysis Corporation, McLean, Virginia (1968).

[23] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM Journal on Applied Mathematics* 11, No. 2 (1963) 431–441.

[24] Y. Bard, "Comparison of gradient methods for the solution of nonlinear parameter estimation problems," *SIAM Journal on Numerical Analysis* 7, No. 1 (1970) 157–186.

[25] R. Fletcher, "An efficient, globally convergent, algorithm for unconstrained and linearly constrained optimization problems," Technical Paper 431, Theoretical Physics Division, Atomic Energy Research Establishment, Harwell, England, *Mathematical Programming*, to appear.