Mathematical Programming 1 (1971) 217–238. North-Holland Publishing Company

PENALTY FUNCTION VERSUS NON-PENALTY FUNCTION METHODS FOR CONSTRAINED NONLINEAR PROGRAMMING PROBLEMS *

Garth P. McCORMICK **
Research Analysis Corporation, McLean, Virginia, U.S.A.

Received 5 November 1970
Revised manuscript received 27 February 1971

The relative merits of using sequential unconstrained methods for solving: minimize $f(x)$ subject to $g_i(x) \geqslant 0$, $i = 1, ..., m$, $h_j(x) = 0$, $j = 1, ..., p$ versus methods which handle the constraints directly are explored. Nonlinearly constrained problems are emphasized. Both classes of methods are analyzed as to parameter selection requirements, convergence to first and second-order Kuhn-Tucker Points, rate of convergence, matrix conditioning problems and computations required.

1. Introduction

The purpose of this paper is to explore the advantages and disadvantages of solving the problem (in E^n)

$$\text{minimize} \quad f(x) \tag{1}$$

$$\text{subject to} \quad g_i(x) \geqslant 0\,, \qquad i = 1, ..., m\,, \tag{2}$$

$$h_j(x) = 0\,, \qquad j = 1, ..., p\,, \tag{3}$$

using sequential unconstrained methods (or penalty function methods) as opposed to the use of algorithms which handle the constraints directly. The particular interest of this paper is to analyze the situation when the constraints (2) and (3) are *nonlinear*. When some of the constraints are linear, a modified method which treats those constraints directly, and not as part of a sequential unconstrained procedure, is to

* This paper was presented at the 7th Mathematical Programming Symposium 1970, The Hague, The Netherlands.

** Now with The George Washington University, Washington, D.C., U.S.A.

be preferred. For simplicity of presentation then we assume that the constraints are nonlinear.

Definition 1

A *sequential unconstrained method* for solving problem A is one which associated with A a sequence of problems

$$\text{minimize} \qquad V^k(f(x), g(x), h(x)) \qquad (A^k)$$

$k = 1, \ldots$, where the solution of each A^k occurs at some point $(s)\, x^k$ which is a relative *unconstrained* minimizer to the function $V^k(f(x), g(x), h(x))$. The functions are constructed so that as $k \to \infty$, a convergent sequence of unconstrained minimizers approaches a constrained minimizer of problem A.

Example

One widely used V^k function is the logarithmic-quadratic loss function (LQ) given as

$$\mathrm{LQ}(x, r_k) = f(x) - r_k \sum_{i=1}^{m} \ln g_i(x) + \sum_{j=1}^{p} h_j^2(x)/r_k$$

where the $\{r_k\}$ have the property that they decrease strictly to zero. To use this function it is required to start with a point x^0 which is in the strict feasible region, i.e.,

$$x^0 \in R^0 \equiv \{x \mid g_i(x) > 0,\ i = 1, \ldots, m\} .$$

Since the function $\mathrm{LQ}(x, r_k)$ is to be minimized and since at the boundary of R^0 $\mathrm{LQ}(x, r_k)$ takes on the value of $+\infty$, this guarantees that all points generated will remain in R^0. I.e., no boundary or infeasible points (with respect to the inequality constraints (2)) will be generated.

The function LQ is called a mixed interior-exterior point penalty function. If there were no equality constraints of the form (3), the LQ function would just have the form

$$f(x) - r_k \sum_{i=1}^{m} \ln g_i(x)$$

and would be called an interior point penalty function. If the constraints (2) were omitted, the LQ function would have the form

$$f(x) + \sum_{i=1}^{p} h_j^2(x)/r_k$$

and would be called an exterior point penalty function. Although it is possible to use an exterior point penalty function to handle problems with inequality constraints, equality constraints can only be handled by the use of exterior point algorithms (so-called because they allow the generation of points not feasible to the constraints, forcing convergence to a feasible point in the limit as $k \to \infty$).

The use of sequential unconstrained methods has a long history. In this paper we will concentrate on some of the newer developments. For history and list of previous work in the area see Fiacco and McCormick [19] Chapter 1, section 1.2, and the references there.

Before talking about the convergence properties of the different algorithms applied to problem A, it is necessary to make some assumptions about the problem. Our basic assumption is that the gradients of constraints binding at any point constitute a linearly independent set.

Linear independence assumption

Let x^0 be a point satisfying (2) and (3). Let $B^0 = \{i | g_i(x^0) = 0\}$. Then $\{\nabla g_i\}, i \in B^0$, $\{\nabla h_j(x^0)\}$ all j is a linearly independent set.

Even the simplex method for linear programming must be modified to take into account the possibility that the linear independence assumption does not hold. We assume it here for simplicity. From this assumption the following characterizations of local and global minimizers to problem A can be stated.

First-order necessary conditions

If the problem functions f, $\{g_i\}$, $\{h_i\}$ are continuously differentiable, then a necessary condition that a point $\bar{x}$ satisfying the constraints (2) and (3) be a local minimizer to problem A is that there exist multipliers $(\bar{u}, \bar{w})$ such that

$$\nabla f(\bar{x}) - \sum_{i=1}^{m} \bar{u}_i \nabla g_i(\bar{x}) + \sum_{j=1}^{p} \bar{w}_j \nabla h_j(\bar{x}) = 0 \tag{4}$$

$$\bar{u}_i g_i(\bar{x}) = 0\,, \qquad i = 1, \ldots, m\,, \tag{5}$$

$$\bar{u}_i \geqslant 0\,, \qquad i = 1, \ldots, m\,. \tag{6}$$

These are sometimes called the Kuhn-Tucker optimality conditions [35].

Second-order necessary conditions

If the problem functions f, $\{g_i\}$, $\{h_j\}$ are twice continuously differentiable, then for every set of multipliers satisfying (4)–(6),

$$z'[\nabla^2 f(\bar{x}) - \sum_{i=1}^{m} \bar{u}_i \nabla^2 g_i(\bar{x}) + \sum_{j=1}^{p} \bar{w}_j \nabla^2 h_j(\bar{x})]z \geqslant 0 \tag{7}$$

for all z where

$$z' \nabla g_i(\bar{x}) = 0 , \qquad i \in \bar{B} \equiv \{i | g_i(\bar{x}) = 0\} ,$$
$$z' \nabla h_j(\bar{x}) = 0 , \qquad j = 1, ..., p .$$

must hold.

One of the ways in which the convergence of an algorithm can be analyzed is whether or not it guarantees convergence to a point satisfying the first-order, or the first- and second-order necessary conditions. We will use the terms first-order convergence and second-order convergence to distinguish these properties.

Of much importance in analyzing the *rate of convergence* of an algorithm to an isolated local minimizer are the following conditions which ensure that a point be an isolated local minimizer.

Second-order sufficiency conditions

A sufficient condition that a point x^* satisfying (2) and (3) be an *isolated* local minimizer to problem A is that there exist multipliers u^*, w^* satisfying (4) – (6) and such that

$$z'[\nabla^2 f(x^*) - \sum_{i=1}^{m} u_i^* \nabla^2 g_i(x^*) + \sum_{j=1}^{p} w_j^* \nabla^2 h_j(x^*)]z > 0 \tag{8}$$

for all z such that $(z \neq 0)$,

$$z' \nabla g_i(x^*) = 0 , \qquad \text{all } i \text{ where } u_i^* > 0$$
$$z' \nabla g_i(x^*) \geqslant 0 , \qquad \text{all } i \text{ where } g_i(x^*) = 0, u_i^* = 0,$$
$$z' \nabla h_j(x^*) = 0 , \qquad j = 1, ..., p .$$

We note that these conditions are slightly stronger than the second-order necessary conditions so that if convergence to a second-order KTP is guaranteed by an algorithm, it is not unreasonable to expect that the second-order sufficiency conditions hold there.

One form of degeneracy that can occur at a local minimizer is the possibility that the multiplier associated with a binding inequality constraint can be zero.

Strict complementary slackness

The term strict complementary slackness is said to apply to a point x^0 satisfying (4)–(6) if $u_i^0 > 0$ whenever $g_i(x^0) = 0$.

We note that the linear independence assumption implies that the set $R^0 \cap Q$ is nonempty, and that the closure of $R^0 \cap Q$ is $R \cap Q$. (Here $Q = \{x \mid h_j(x) = 0, j = 1, ..., p\}$, $R = \{x \mid g_i(x) \geqslant 0, i = 1, ..., m\}$.

If the problem functions f, $\{-g_i\}$ are convex functions, and the functions $\{h_j\}$ are linear, then the first order conditions are *sufficient* that $\bar{x}$ be *global* minimizer of the problem A. The second order necessary conditions add no additional information since the matrix

$$\nabla^2 f(\bar{x}) - \sum_{i=1}^{m} \bar{u}_i \nabla^2 g_i(\bar{x})$$

is positive semi-definite. Because of this, the second-order strict sufficiency conditions are often overlooked. However, since they guarantee the uniqueness of the minimizer, they are still needed to assure a decent *rate* of convergence.

Some algorithms for nonlinearly constrained (e.g. cutting plane methods) problems explicitly assume that the functions $\{-g_i\}$ are convex. This severely limits the applicability of these methods at the present time, but I feel that in the future all programming problems will be solved by solving a sequence of "convex approximating" problems so this limitation will not be as important [46].

Near a point where the second-order sufficiency conditions hold, the penalty function method reduces to finding the unconstrained minimizer of a strictly convex function. Thus one is essentially solving a convex (unconstrained) programming problem. This advantage is not shared by many methods.

Another desirable property that penalty function methods have is that a convex programming problem is converted to a sequence of unconstrained convex minimization problems. Thus *convergence* to the global solution of a convex programming problem is guaranteed.

2. One obvious advantage of penalty functions

The original motivation (at least for the work of Fiacco and McCormick) for using the penalty function approach was that it bypassed the necessity for moving along the boundary of a nonlinear constraint. Several methods based on such motion met in failure. Two different approaches to this problem which have had limited success are summarized. Consider the case for inequality constraints only.

The first approach for maintaining feasibility with respect to inequality constraints is essentially that suggested by Zoutendijk [67]. At a boundary point of a nonlinearly constrained region, the next direction of search is required to form slightly less than a right angle with the gradient(s) of the active constraint(s). (See fig. 1.) This direction vector is also required to effect a small decrease in the value of the objective function if movement a short distance along that vector is made. Thus the existence of a non-empty strict feasible region is a necessary requirement of this method of coping with nonlinear constraints. Very sophisticated rules for determining the angle the new direction makes with the constraint gradient can be made. The choice of angle also can be made via an optimization problem. Using this approach it becomes immediately obvious that each active constraint requires individual attention. The usual difficulty encountered in using this approach is

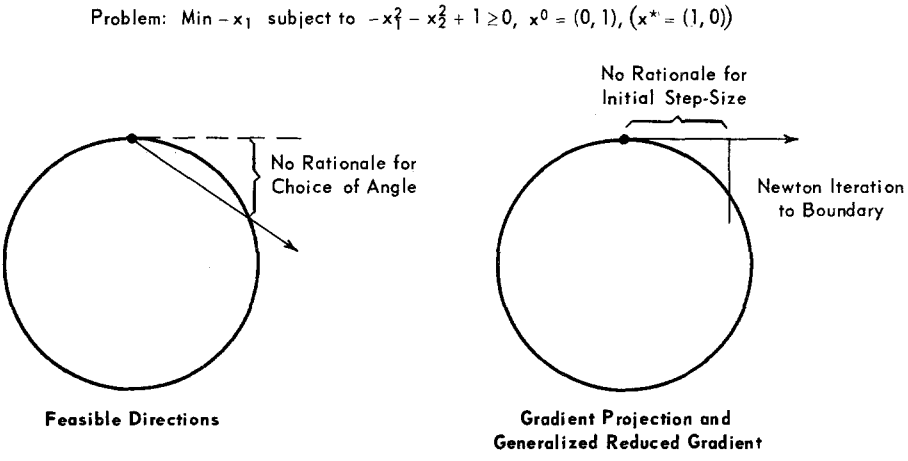

Fig. 1. Handling of nonlinear constraints.

that many, many small steps must be made bumping into the boundary, moving interior, reencountering the old boundary, etc. It is significant that today there is no generally available code which implements this concept. The code used to solve some of the test problems in the study by Colville [12] to implement the ideas of feasible directions handled only linear constraints.

The reason for the failure of this approach is the lack of concern (implicitly or explicitly) with the curvature of the active constraint(s) at the boundary. Current work by Graves and Whinston [25] implicitly tries to take this into account.

The second general approach was suggested by Rosen [59] in his extension of the gradient projection method to handle nonlinear constraints and is in essence the same approach used later by Abadie et al. [3] in the Generalized Reduced Gradient scheme (GRG).

The idea here is to generate a direction of search (see fig. 1) treating the constraints as though they were linear, and going to a point on that vector. In general, particularly if the constraint set is convex, any point on the vector away from the starting vector will be infeasible. A second move, back to the boundary of the feasible region, is then made by a modified form of Newton's method of solving nonlinear equations. (The two methods differ basically in that one uses the projection form of the constraint gradients, the other the variable elimination form. Also, the choices for how far to go along the initial vector, how far to cut back, etc., are more elaborate in the generalized reduced gradient method.) The ideas in Wilde and Beightler [62] are similar but the problem of feasibility is not considered by them. This idea has been coded and, at least for the GRG, does solve nonlinearly constrained problems (Abadie and Carpentier [2], Colville [12], and Guigou [29]). More will be said about this method in the section on parameter selection.

This idea, a two-phased move for each iteration, is essentially a way of moving along the nonlinear arc describing the boundary of the feasible region. In principle it could be used to solve equality constrained problems if one took the effort to ensure that the second phase converged to the boundary. I do not know if it has been tried on these kinds of problems.

My own preference for a way to handle the curvature of the nonlinear constraints is a modification of the two-phased method just described. This modification is contained in McCormick [44].

In summary, the penalty function approach, as opposed to the techniques just described, avoids separate consideration of moving along the boundary and has the advantage of simplicity. To solve problems with the penalty function approach, one needs only to consider methods of minimizing unconstrained functions (possibly subject to linear constraints).

3. A concomitant disadvantage

The literature on methods of minimizing unconstrained functions has grown exponentially in the last few years. Application of these techniques to the minimization of penalty functions has shed some light on the rate of convergence properties of these methods and on the difficulties of minimizing penalty functions.

The essential problem, first pointed out by Murray [50] and elaborated on by Lootsma [42] is that as the parameters which define the penalty function are adjusted to allow the unconstrained solutions to approach the constrained solutions, invariably the Hessian matrices of the penalty functions become ill-conditioned. Since acceleration of convergence of these algorithms depends in general on the closeness to which the function being minimized approximates a positive definite quadratic form and the condition number of the Hessian matrix, (although there is not much in the literature on this point) these techniques often fail to satisfactorily minimize penalty functions. Since many of these techniques, particularly those of the conjugate directions variety, require a "line search" or one dimensional minimization, the usual quadratic fit in one variable which makes these methods work for unconstrained nonpenalty function problems, does not really apply. This can be seen by examining eq. (8) which gives the explicit formulas for the Hessian matrix of the penalty function $LQ(x,r_k)$,

$$\begin{aligned}
&\nabla^2(f(x) - r_k \sum \ln g_i(x) + \sum h_j^2(x)/r_k) \\
&= \nabla^2 f(x) - \sum [r_k/g_i(x)]\nabla^2 g_i(x) + \sum [2h_j(x)/r_k]\nabla^2 h_j(x) \\
&\quad + \sum \nabla g_i(x)[r_k/g_i^2(x)]\nabla' g_i(x) + \sum \nabla h_j(x)[2/r_k]\nabla' h_j(x)\,. \qquad (8)
\end{aligned}$$

The scalars $r_k/g_i[x(r_k)]$ and $2h[x(r_k)]/r_k$ tend to the multipliers associated with the optimality conditions. Thus the first three terms of (8) tend to the Hessian matrix of the Lagrangian function (see eq. (7)).

This has (using the linear independence assumption) finite eigenvalues. The last three terms, involving the outer product of the gradients of the problem functions, have associated scalars $r_k/g_i(x)^2$ and $2/r_k$ which (for the inequality constraints with $u_i^* > 0$) tend to increase without bound. This portion of the Hessian matrix of the penalty function yields infinite eigenvalues and thus provides the ill-conditioning problem. (The one exception to this is the case where there are n binding constraints at the local minimizer converged to, and the constraint gradients there are linearly independent.)

For the inequality constraints, when an unconstrained minimizer is achieved and the value of r is changed from r_k to r_{k+1} where $r_k > r_{k+1}$, there is an initial reduction of the eigenvalues associated with the inequality constraints since $r_{k+1}/g_i[x(r_k)]^2 < r_k/g_i[x(r_k)]^2$. Any eigenvalue associated with the equality constraints is increased however as $2/r_{k+1} > 2/r_k$. This is why the difficulty is more serious when dealing with exterior point methods than with interior point methods. In recent months, several papers have proposed algorithms to deal with these problems (Fletcher and McCann [22], Murray [49], Powell [56], Pietrzykowski [55], and Fletcher [20]). It is yet too early to evaluate the efficacy of these proposals.

This problem is not as serious if a second-order method is used to minimize the unconstrained function and if the theory of extrapolating from one minimizer to another is used. These matters are dealt with at some length in the book by Fiacco and McCormick [19]. Because of the reluctance of most people to consider using second derivatives, these techniques are not commonly used. For problems where second derivatives are readily available, the conditioning problem is not as crucial for *interior* point methods. Where *exterior* point algorithms are used (for example when one has equality constraints) the results are not as satisfying.

4. Parameter selection

One desirable feature of an algorithm, as I see it, is to be "parameter free". That is, in order for the algorithm to converge, or have a decent rate of convergence, there should not have to be arbitrary choices about parameter values. Sometimes these choices are incorporated into the computer programs, sometimes they are left to the user. On one

occasion I sat in an audience where new results of someone's algorithm concerning performance on some of the problems in Colville [12] were called in from the floor. In attempting to solve the problems, people adjusted this and that parameter and eventually got their solution times down. In many of the algorithms which are applied to engineering design problems, such as the simplex search method [52] or the complex method [8], one is confronted with alphas and betas and suggested gamma values.

Methods for minimizing unconstrained functions are often (but not always) parameter free. For the method of steepest descent and methods of conjugate directions, the optimum along the search vector is taken. For the Newton, or some of the new quasi-Newton methods, a step-size of one is prescribed.

The distinction should be made between "tolerances" and "parameters", There will always be a certain amount of trouble with determining linear dependence of vectors, constraint satisfaction and, in general, what is a good approximation to zero. Tolerances are not my concern here. What is important is to have a good rationale for the epsilon of the method of feasible directions, and the factors of 10 and 100 by which the initial vector of the GRG is reduced.

There are few methods for handling nonlinear constraints which do not depend upon parameter selection. For penalty function methods the parameter choices required are obvious. There are, however, versions of penalty functions which do not require parameter selection. For the interior case see Huard's method of centers [32], and a particular interior point parameterless function by Fiacco and McCormick, Chapter 7 in [19]. For exterior point parameterless methods see Fiacco [18] pp. 85–93, Morrison [47], and Kowalik et al. [34]. In these methods the sequence of iterations is determined by information generated only by the algorithm.

Recent work by Murray [49], Powell [56], and Roode [58] has touched on this point. Also, work by Greenberg and Pierskalla [26] where all the constraints are collapsed into one might eliminate, in a computationally efficient way, dependence on parameter selection.

In Fiacco and McCormick [19] reasonable criteria for initial parameter selection are developed. My own feeling is that it is possible to make parameter selection in penalty function techniques an automatic technique with reasonable choices satisfying the desirable characteristics that any algorithm should have. (One such criterion is that if the

initial starting point is close to the optimizer, the algorithm sho ıld not generate points which are farther away – sometimes parameter selection in penalty function methods fails to satisfy this desirable characteristic.) In the end, however, the work involved justifies development of a nonpenalty function algorithm.

From my own experience, the particular method of centers developed in Fiacco and McCormick [19] has not produced better results than the regular penalty function approach.

5. Closeness to linear programming methods

One way to view methods for handling nonlinear constraints is to examine the formulas used by the algorithms to see if, in the case the constraint functions were actually linear and the objective function were linear, the methods would find the solution in a finite number of iterations. This criterion was suggested by Wolfe in Fletcher [21] p. 201. The gradient projection method has this property. Although there is no published proof, the generalized reduced gradient method probably does also. The method of feasible directions * does not because it, along with the ricochet method of Greenstadt [28], would move interior to the constraint if it were linear. (The ricochet method can probably be modified to satisfy this property and perhaps, as outlined in [28], does do this in the final stages of the algorithm.)

On the surface, penalty functions seem to be terribly inefficient since they do not have this desirable property. A case can be made, however, for the manner in which penalty functions use linear constraints. One must, however, look at the method for minimizing an unconstrained function and assume that it is the generalized Newton method. Consider the problem

$$\begin{aligned} &\text{minimize} && c'z \\ &\text{subject to} && Ax = b \qquad (A\ n \times n \text{ and invertible}). \end{aligned} \tag{9}$$

* Note that this is a different statement from saying that the method of feasible directions does not minimize a linear function in a finite number of steps. As described in [66] it is a finite procedure for minimizing a positive semidefinite quadratic form subject to linear constraints. The point here is that its use of nonlinear constraints does not reduce to how linear constraints are explicitly handled by the method.

Suppose the logarithmic quadratic loss function is applied to this problem. The function to be minimized for any $r_k > 0$ is

$$\text{minimize} \qquad \mathrm{LQ}(x, r_k) = c'x + \|Ax - b\|^2 / r_k \,. \tag{10}$$

Let x^0 be the starting point. At x^0,

$$\nabla L(x^0, r_k) = c + A'(Ax^0 - b)(2/r_k) \,,$$

$$\nabla^2 L(x^0, r_k) = A'A(2/r_k) \,.$$

The unconstrained minimizer is found in one Newton move as

$$x(r_k) = x^0 - [A'A]^{-1} [c + A'(Ax^0 - b)(2/r_k)] (r_k/2) \,. \tag{11}$$

In order to obtain an unconstrained minimizer then requires of the order of n^3 operations. Using the theory of extrapolation (see Fiacco and McCormick [19], chapters 5 and 8), an estimate of the solution can be made once the unconstrained minimizer is available. Using a one term Taylor series, expansion yields

$$x^* = x(r_k) - r_k \, \mathrm{d}x(r_k)/\mathrm{d}r \,. \tag{12}$$

The term $\mathrm{d}x(r_k)/\mathrm{d}r$ is already available and for no more multiplications and additions the exact solution to (9) is obtained.

Thus the solution to this trivial problem requires of the same order of operations (n^3) as that required by the simplex method.

A more general statement of the efficiencies in minimizing the penalty function when the constraints are of a special form (linear, separable, factorable) is contained in Fiacco and McCormick [19], chapter 8, section 8.3. Basically, the idea is to update the inverse to the Hessian matrix of the penalty function using the techniques akin to those developed for the gradient projection method of Rosen. This keeps the order of operations to change from one Hessian inverse to another at the order of n^2.

Wolfe's criterion in another form is that algorithms, when they can, should use information from iteration to iteration and reduce the order of operations, multiplications and additions, required to solve the problem. It is the failure to meet this criterion that makes suspicious such techniques as the complex method of Box [8], the search methods of Hook and Jeeves [31], and other engineering design oriented methods. It also rules out such techniques as random methods suggested by several Russian authors – Mátyáš [43] and Motkus [48].

6. Convergence of the algorithms

To say that an algorithm "converges" is to say nothing without an explicit definition of the characteristics of the point converged to. Generally speaking, when people say an algorithm converges they mean that cluster points of the sequence generated by the algorithm satisfy the Kuhn-Tucker first-order necessary conditions. For convex programming problems these conditions are sufficient to ensure that a point be a global minimizer and thus for convex problems it is usually enough to prove the first-order conditions are satisfied. One well known algorithm, the cutting plane method (Cheney and Goldstein [11], Kelley [33], Eaves and Zangwill [15], Levitin and Polyak [40]) can be proven to be a convergence algorithm for the convex case *.

It is possible, with enough rules, to guarantee convergence of just about any algorithm to a KTP. A general theory of this has been established in Zangwill [65]. Although for many of the methods mentioned in this review no convergence proofs are published, one can, with the proper anti-zig-zagging precautions, prove that they converge to a first-order KTP.

Until recently, very little has been done about second-order convergence. This probably derives from the reluctance of people to use algorithms which require second derivatives. For penalty functions however, since the use of the minimizer of the unconstrained problem for each k implies the Hessian matrix is positive semi-definite, it is easy to show (see Fiacco and McCormick [19]) that cluster points satisfy the first- and second-order necessary conditions. The most useless, and the most used phrase one sees, is "the algorithm converges locally".

Convergence theorems for penalty functions are relatively easy to come by and appear in many places. Unfortunately in recent years, although sophistication in providing convergence has increased, it has often been at the expense of an understanding of the importance of the *rate of convergence* of an algorithm.

* One difficulty with the cutting plane algorithm is that artificial constraints must be placed on the problem to ensure that the linear programming problems generated by the algorithm have finite solutions. For example, in the problem of fig. 1, the first linear programming problem generated would be, using $x^0 = (0, 1)'$ as the starting point, minimize $-x_1$ subject to $2x_2 - 2 \leqslant 0$. This problem has no solution. One criterion for a good algorithm is that the algorithm should generate only bounded sequences of points if the solution set of the original problem is bounded.

7. Rate of convergence

For the unconstrained problem, the rate of convergence of various algorithms which have been around for 10 years or more is just now being understood. Generally, the extension of these rates to problems with linear constraints is a simple matter. One merely looks at the function defined in the subspace of binding constraints and regards the problem unconstrained in a linear subspace of smaller dimension. For nonlinear constraints, however, the same notions do not apply. It is easy to see this by thinking of the minimization of a linear function on the boundary of a "strictly convex" set.

The concept of rate of convergence has been extensively explored in the area of the solution of nonlinear equations. These results can have a direct application to the field of nonlinear programming if one looks at the problem of trying to solve the Kuhn-Tucker conditions (4) – (6) by the use of Newton's method. Suppose one is at a point x with an estimate of the Lagrange multipliers u and w. Then Newton's method as applied to this set of equations would yield the iteration,

$$\begin{bmatrix} x^{k+1} \\ u^{k+1} \\ w^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ u^k \\ w^k \end{bmatrix} - \begin{bmatrix} \nabla^2 \mathcal{L}, -\nabla g_1, \ldots, -\nabla g_m, \nabla h_1, \ldots, \nabla h_p \\ u_1 \nabla' g_1 \quad g_1 \qquad\qquad 0 \\ \vdots \qquad\qquad \ddots \\ u_m \nabla' g_m \qquad g_m \\ \nabla' h_1 \\ \vdots \qquad 0 \qquad\qquad 0 \\ \nabla' h_p \end{bmatrix}^{-1} \begin{bmatrix} \nabla \mathcal{L} \\ u_1 g_1 \\ \vdots \\ u_m g_m \\ h_1 \\ \vdots \\ h_p \end{bmatrix} \quad (13)$$

(evaluated at (x^k, u^k, w^k))

where

$$\mathcal{L}(x, u, w) \equiv f(x) - \sum_{i=1}^{m} u_i g_i(x) + \sum_{j=1}^{p} w_j h_j(x) . \quad (14)$$

The usual theorems about the convergence and rate of convergence of Newton's method would apply here. Note the obvious necessity of using the second derivatives of the objective function and constraints.

Now for the iterations to converge, and have a decent rate of convergence, the determinant of the matrix in (13) must be nonzero. It is easy to show that in a neighorhood of a point (x^*,u^*,w^*) satisfying the second-order sufficiency conditions and strict complementary slackness, the conditions necessary for the convergence of (13) hold. The implications of the above statement are too many to be examined here. Approaches which can be analyzed from this point of view have been suggested by Graves and Whinston [25], Wilson [63], Bard and Greenstadt [6], Broyden and Hart [9], and Kwakernaak and Strijbos [37].

Certainly one would hope that once the constraints active at the solution were identified by an algorithm, any iteration would in some way approximate that given by eq. (13). Other requirements might be desirable, for example (ignoring the equality constraints), feasibility and decreasing objective function values are usual requirements for a nonlinear programming algorithm.

An interesting and computationally important acceleration technique for penalty functions based on extrapolation of the minimizing trajectory in a Taylor's series expansion is the by-product of convergence of the algorithm to a point satisfying the second-order sufficiency conditions. This is discussed in detail in [19], chapters 5 and 8.

It is its slow *rate of convergence* that I feel has prevented the cutting plane approach and the tangential approximation method of Hartley and Hocking [30] from being widely used and successful. This is partly due to the increase in the numbers of linear constraints generated and partly due to the nearly linear dependence of the vectors defining the linear constraints. Some of the newer methods (Lions and Teman [41], Laurent and Martinet [38]) of the cutting plane variety may eliminate the problem of increased numbers of constraints, but the problem of linear dependence remains.

Although the scheme suggested by Rosen for handling the nonlinear constraints does not depend upon the choice of the initial direction vector, it is usually the case that a choice analogous to the steepest descent direction of Cauchy for the unconstrained problem is used. In some versions of the reduced gradient code (for linear constraints only) the direction of search is analogous to the choice suggested by Fletcher and Reeves – a variation of the conjugate gradient method. This modification is important for the linear case in accelerating the *rate* at which the method converges to an unconstrained minimizer in the

unconstrained case. In none of the schemes, feasible directions, generalized reduced gradient, or gradient projection has any attention been paid to the second-order sufficiency conditions for nonlinear constraints. Hence nothing can be said about their *rate* of convergence.

8. Miscellaneous topics – special structures

In recent years much work has been done in the area of geometric programming. The geometric programming problem (Duffin et al. [14]) can be written

$$\underset{x}{\text{minimize}} \quad f(x) = \sum_{l=1}^{q_0} c_{0l} \prod_{j=1}^{n} x_j^{a_{0,l,j}} \qquad \text{Primal}$$

$$\text{subject to} \quad g_i(x) = 1 - \sum_{l=1}^{q_i} c_{il} \prod_{j=1}^{n} x_j^{a_{i,l,j}} \geqslant 0\,, \quad i = 1,...,m, \qquad (15)$$

$$x_j > 0\,, \quad j = 1, ..., n\,.$$

It is required that the parameters $\{c_{i,l}\}$ be nonnegative.

I think there is a wide confusion between the theory of geometric programming, which is quite elegant and interesting, and algorithms for *solving* geometric programming problems. The theory of geometric programming is essentially a development of duality theorems, i.e., there is a dual to the primal programming problem given above as

$$\underset{\delta_{il}}{\text{maximize}} \quad v(\delta) = \left[\prod_{i=0}^{m} \left(\prod_{l=1}^{q_i} (c_{il}/\delta_{il})^{\delta_{il}}\right)\right] \left[\prod_{i=1}^{m} \left(\sum_{l=1}^{q_i} \delta_{il}\right)^{\left(\sum_{l=1}^{q_i} \delta_{il}\right)}\right]$$

$$\text{subject to} \quad \delta_{il} \geqslant 0\,, \quad (l = 1, ..., q_i)\,, \quad i = 0, ..., m\,, \qquad \text{Dual}$$

$$\sum_{l=1}^{q_0} \delta_{0l} = 1 \qquad (16)$$

$$\sum_{i=0}^{m} \sum_{l=1}^{q_i} (a_{i,l,j})(\delta_{i,l}) = 0\,, \quad j = 1, ..., n\,.$$

The dual problem is equivalent to the maximization of a concave function subject to linear inequality and equality constraints. (This is

true since the logarithm of the function $v(\delta)$ is concave.) To solve a geometric programming problem, one needs only to solve the dual, and the optimum primal variables can be obtained from well-known relations.

There are some codes for solving the dual, but it should be obvious that any algorithm for solving linearly constrained problems can be applied to solving (16). This, to my knowledge, has not been emphasized or attempted. (Of course one could also convert the original primal problem to a convex programming problem by a transformation of the variables (Charnes [10]). Newer work of Avriel and Williams [5], Duffin and Peterson [13], and Passy [54] is aimed at by-passing some of the restrictions of the current formulation of geometric programming problems. A code for solving the dual has been programmed by Woolsey [64] using the techniques of separable programming and decomposition.

Method of Lagrange multipliers

Another confusion of longer history is the so-called "method of Lagrange (or undetermined) multipliers". What is usually meant is the method "implied" by writing down the first-order necessary conditions which hold at a local minimizer. And, usually, an equality constrained problem is considered. Sometimes one can, for small problems, write dowm the necessary conditions and by a series of tricks solve the problem. But there is no natural algorithm for doing this.

There is a method called the Generalized Lagrange Multiplier Technique (GLMT) (Everett [15]). However, there is no published algorithm in the sense of an explicit procedure for generating points iteratively which converge to a solution. (As a corollary, there are no convergence or rate of convergence proofs.) Finally, there is no code generally available which handles nonlinear programming problems. (See Fiacco and McCormick [19] for a fuller discussion.)

For convex programming problems there is a Lagrange multiplier method (essentially an elaboration of a duality theorem) which has an elegant and firm theoretical basis (Falk [17]). Methods for solving the equivalent dual problem have been coded and this has worked successfully.

Separable programming

Many nonlinear programming problems are very nearly linear and the

nonlinearities can be expressed as sums of functions of single variables. Special techniques to handle these kinds of problems have been proposed and, even better, incorporated into large linear programming computer packages (see Beale [7]).

Special structures

Except for problems in geometric, linear, quadratic, or separable format, not much has been done in analyzing nonlinear functional forms. But in fact, a general class of problems called "factorable problems" is in use most of the time. A method for manipulating these functions (taking their derivatives first and second – automatically) is implemented in a new programming language (Pugh [57]). Using factorable functions it is possible to obtain global solutions to not necessarily convex programming problems.

10. Conclusion

None of the methods included in this survey satisfies all the criteria proposed for a good algorithm for handling nonlinear constraints. Each of the algorithms has its own motivation, and for the types of problems it was motivated to solve, does very well. Perhaps it is too much to hope that one algorithm will contain enough flexibility to do well on the variety of problems and structures that arise when constraint functions are allowed to take on any type of nonlinearities.

If there is a major breakthrough, it will probably come from looking at nonlinearly constrained problems from an algebraic as well as a geometric point of view. What is required is a new synthesis of what a nonlinear function is in algebraic terms. It is required to distinguish between linear and nonlinear, convex and nonconvex functions in a computer implementable way. To do this, people someday are going to have to examine second derivatives of nonlinear functions because that is what makes all these distinctions possible.

References

[1] J. Abadie, (Ed.), *Nonlinear programming* (North-Holland Publishing Company, Amsterdam, 1967).

[2] J. Abadie and J. Carpentier, "Some numerical experiments with the GRG method for nonlinear programming," Paper HR 7422, Electricité de France, 1967.

[3] J. Abadie and J. Carpentier, "Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints," in: *Optimization,* Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 37–47.

[4] A. Auslender, "Méthodes du second ordre dans les problèmes d'optimisation avec constraintes," *Revue Française d'Informatique et de Recherche Opérationelle,* No. R-2 (March-April, 1969).

[5] M. Avriel and A.C. Williams (unpublished manuscript).

[6] Y. Bard and J.L. Greenstadt, "A modified Newton method for optimization with equality constraints," in: *Optimization*, Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 299–306.

[7] E.M.L. Beale, *Mathematical programming in practice* (Sir Isaac Pitman & Sons, Ltd., London, 1968).

[8] M.J. Box, "A new method of constrained optimization and a comparison with other methods," *Computer Journal* 8 (1965).

[9] C.G. Broyden and W.E. Hart, "A new algorithm for unconstrained optimization," Paper presented at the Seventh Mathematical Programming Symposium, The Hague, the Netherlands, Sep. 1970.

[10] A. Charnes and W.W. Cooper, "Optimizing engineering designs under inequality constraints," *Cahiers Centre Etudes Recherche Opérationelle* 6 (1964) 181–187.

[11] E.W. Cheney and A.A. Goldstein, "Newton's method for convex programming and Tchebycheff approximation," *Numer. Math.*, 1 (1959) 253–268.

[12] A.R. Colville, "A comparative study on nonlinear programming codes," IBM New York Scientific Center Report, 320–2949, 1968 (revised, June 1970).

[13] R.J. Duffin and E.L. Peterson (unpublished manuscript).

[14] R.J. Duffin, E.L. Peterson and C. Zener, *Geometric Programming – Theory and Application* (John Wiley & Sons, Inc., New York, 1967).

[15] B.C. Eaves and W.I. Zangwill, "Generalized cutting plane algorithms," Working Paper No. 274, Center for Research in Management Science, University of California, Berkeley, California, July 1969.

[16] H. Everett, "Generalized Lagrange multipliers method for solving problems of optimum allocation of resources," *Operations Research* 11: 399–417 (1963).

[17] J.E. Falk, "Lagrange multipliers and nonlinear programming," *J. Math. Anal. Appl.*, 19(1) (July 1967).

[18] A.V. Fiacco, "Sequential unconstrained minimization techniques," Ph.D. Dissertation, Northwestern University, Evanston, Ill., (June 1967).

[19] A.V. Fiacco and G.P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques* (John Wiley & Sons, Inc., New York, 1968).

[20] R. Fletcher, "A class of methods for nonlinear programming with termination and convergence properties," T.P. 368, A.E.R.E., Harwell, England (May 1969).

[21] R. Fletcher. (Ed.), *Optimization* (Academic Press, London and New York, 1969).

[22] R. Fletcher and A.P. McCann, "Acceleration techniques for nonlinear programming," in *Optimization*, Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 203–246.

[23] F.J. Gould, "Extensions of Lagrange multipliers in nonlinear programming," *SIAM Journal of Applied Mathematics* 17 (1969) 1280–1297.
[24] G.W. Graves and A.B. Whinston, "Application of mathematical programming to regional water quality management," Paper presented at the Seventh Mathematical Programming Symposium, The Hague, the Netherlands, Sep. 1970.
[25] G.W. Graves and A.B. Whinston, "The application of a nonlinear programming algorithm to a second order representation of the problem," Working Paper No. 108, Western Management Science Institute, University of California, Los Angeles, California, 1966.
[26] H.J. Greenberg and W.P. Pierskalla, "Surrogate mathematical programming," *Operations Research* (to appear).
[27] H.J. Greenberg, "The generalized penalty function – surrogate model," Technical Report CP 70005, Computer Science/Operations Research Center, Southern Methodist University, Dallas, Texas (March 1970).
[28] J.L. Greenstadt, "A ricocheting gradient method for nonlinear optimization," *J. Soc. Ind. Appl. Math.*, 14(3) (1966).
[29] J. Guigou, "Experiences numeriques comparative concernant la methode du gradient conjugue," *Bulletin de la Direction des Etudes et Recherches,* Electricité de France, Série C, No. 2 (1968) pp. 79–92.
[30] H.O. Hartley and R.R. Hocking, "Convex programming by tangential approximation," *Management Science* 9 (1963) 600–612.
[31] R. Hooke and T.A. Jeeves, "Direct search solution of numerical and statistical problems," *J. Assn. Comput. Mach.* 8 (1961).
[32] P. Huard, "Resolution of mathematical programming with nonlinear constraints by the method of centers," in: *Nonlinear Programming,* Ed. J. Abadie (North-Holland Publishing Company, Amsterdam, 1967).
[33] J.E. Kelley, Jr., "The cutting plane method for solving convex programming problems," *J. Soc. Ind. Appl. math.* 8(4) (1960) 703–712.
[34] J. Kowalik, M.R. Osborne and D.M. Ryan, "A new method for constrained optimization problems," *Operations Research* 17(6) (Nov.-Dec. 1969).
[35] H.W. Kuhn and A.W. Tucker, "Non-linear programming," in: *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, Ed. J. Neyman (University of California Press, Berkeley, 1951) pp. 481–493.
[36] H.P. Kunzi and W. Oettli, *Nichtlineare Optimierung: neue Verfahren Bibliographie* (Nonlinear Programming: New Methods Bibliography) (Springer Verlag, Berlin) Lecture Notes in Operations Research and Mathematical Systems, No. 16, 1969.
[37] H. Kwakernaak and R.C.W. Strijbos, "Extremization of functions with equality constraints," Paper presented at the Seventh Mathematical Programming Symposium, The Hague, the Netherlands, Sep. 1970.
[38] P.J. Laurent and B. Martinet, "Methodes duales pour le calcul du minimum d'une fonction convexe sur une intersection de convexes," *Colloque d'Optimisation,* Nice, 30 juin–5 juillet 1969 (Springer Verlag) (to appear).
[39] V.J. Law and R.H. Fariss, "Treatment of inequality constraints in nonlinear programming via rotational discrimination," Paper presented at SIAM National Meeting, June 11–14, 1968.
[40] E.S. Levitin and B.T. Polyak, "Constrained minimization methods," *U.S.S.R. Computational Mathematics and Mathematical Physics*, (English translation), 6(5): 1–50 (Pergamon Press, Ltd., Oxford, 1966).
[41] J.L. Lions and R. Teman, "Une méthode d'eclatement des operateurs et des constraintes en calcul des variations," *C.R. Acad. Sci. Paris* 263 (1966) 563–565.

[42] F.A. Lootsma, "Hessian matrices of penalty functions for solving constrained-optimization problems," Philips Research Reports, Vol. 24, pp. 322–330, Gloeilampenfabrieken-Eindhoven-Netherlands, (Aug. 1969).
[43] J. Mátyáš, "Random optimization," *Automatika i Telemechanica* 26 (1965) 246–253. (in Russian).
[44] G.P. McCormick, "An arc method for nonlinearly constrained programming problems," MRC Technical Summary Report # 1073, Mathematics Research Center, U.S. Army, University of Wisconsin, Madison, Wisconsin, June 1970.
[45] G.P. McCormick, "A second order method for the linearly constrained nonlinear programming problem," *Proceedings of the Symposium on Nonlinear Programming* Mathematics Research Center, U.S. Army, University of Wisconsin, Madison, Wisconsin, May 4–6, 1970.
[46] G.P. McCormick, "Global solutions to optimization problems," unpublished notes.
[47] D.D. Morrison, "Optimization by least squares," *SIAM Journal of Numerical Analysis* 5: (1968) 83–88.
[48] I.B. Motkus, "On a method of distributing random tests in solving many-extremum problems."
[49] W. Murray, "An algorithm for constrained minimization," in: *Optimization*, Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 247–258.
[50] W. Murray, "Behaviour of Hessian matrices of barrier and penalty functions arising in optimization," Ma. 77, National Physical Laboratory, April 1969.
[51] B.A. Murtagh and R.W.H. Sargent, "A constrained minimization method with quadratic convergence," in: *Optimization*, Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 215–346.
[52] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal* 7 (1965) 308.
[53] L.D. Pascual and A. Ben-Israel, "Constrained minimization of posynomials by geometric programming," Series in Applied Mathematics, Report No. 69–3, Northwestern University, Evanston, Ill. (1969).
[54] U. Passy and D.J. Wilde, "Generalized polynomial optimization," *SIAM Journal of Applied Mathematics* 15 (1967) 1344–1356.
[55] T. Pietrzykowski, "An exact potential method for constrained maxima," *SIAM Journal Numerical Analysis* 6(2) (June 1969).
[56] M.J.D. Powell, "A method for nonlinear constraints in minimization problems," in: *Optimization*, Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 283–298.
[57] R.E. Pugh, "A language for nonlinear programming problems," RAC–TP–407, Research Analysis Corporation, McLean, Virginia (1970).
[58] J.D. Roode, "Generalized Lagrangian functions and mathematical programming," in: *Optimization*, Ed. R. Fletcher (Academic Press, London and New York, 1969) pp. 327–338.
[59] J.B. Rosen, "The gradient projection method for nonlinear programming, Part II; nonlinear constraints," *J. Soc. Ind. Appl. Math.* 9 (1961) 514–532.
[60] R.W.H. Sargent and B.A. Murtagh, "Projection methods for nonlinear programming," Paper presented at the Seventh Mathematical Programming Symposium, The Hague, the Netherlands, Sep. 1970.
[61] D.J. Wilde and G. Blau (unpublished manuscript).

[62] D.J. Wilde and C.S. Beightler, *Foundations of optimization* (Prentice-Hall, Englewood, N.J., 1967).
[63] R.B. Wilson, "A simplicial method for convex programming," Dissertation, Harvard University (1963).
[64] R.E.D. Woolsey (private communication).
[65] W.I. Zangwill, *Nonlinear Programming: a unified approach*, Prentice-Hall International Series in Management Science (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1969).
[66] G. Zoutendijk, "Algorithm based on the principle of feasible directions," Paper presented at the Nonlinear Programming Symposium, Madison, Wisconsin, May 4–6, 1970.
[67] G. Zoutendijk, *Methods of feasible directions* (Elsevier Publishing Company, Amsterdam and New York, 1960).