Computer Sciences Department
The University of Wisconsin
1210 West Dayton Street
Madison, Wisconsin 53706

Techniques of Optimization[1)2)]

by

O. L. Mangasarian[3)]

Technical Report #124

May 1971

1) Invited Lecture, International Conference on Design Automation, Toronto, September 8-10, 1971

2) Supported by National Science Foundation Grant GJ-362 and U. S. Army Contract No. DA-31-124-ARO-D-462.

3) Computer Sciences Department and Mathematics Research Center, University of Wisconsin, 1210 West Dayton Street, Madison, Wisconsin 53706

CONTENTS

ABSTRACT

This paper describes computational algorithms for solving unconstrained and constrained optimization problems.

I. INTRODUCTION

Optimization problems occur in engineering, physical and social sciences. Frequent occurrence of such problems and the advent of fast large scale computers has produced, beginning in 1948 [1], a mathematical optimization discipline which has been concerned with the theory and computational algorithms associated with these problems. This paper is an attempt at the description of some of the more important algorithms for solving nonlinear optimization problems. We say nonlinear, because the optimization problem that is completely linear (that is linear objective and constraints) leads to the now-classical simplex algorithm for solving linear programming problems [2, 3].

The theory of mathematical optimization (also referred to as mathematical programming and nonlinear programming) is well established and available in a number of books [4-8]. The algorithms [9-12] of mathematical optimization however are not as well established yet, although there is great activity in the field now. One difficulty is, that unlike the linear programming case, there does not seem to be one algorithm or even a class of algorithms that is universally efficient on nonlinear optimization problems. There have been

attempts at unification of the algorithmic theory [13-14] but these attempts are in their infancy yet and do not for example give criteria for selecting algorithms nor do they give estimates of how fast these algorithms will converge. Also the few comparative computational studies of algorithms are rather limited in scope and have generally tested small problems [15-16]. Roughly speaking however, it can be stated that general nonlinear optimization problems with a few hundred variables and constraints can, by judicious choice of algorithm, be solved today [17].

This paper describes three types of algorithms: one dimensional optimization (two algorithms), unconstrained optimization (three algorithms), and constrained optimization (four algorithms).

II. ONE DIMENSIONAL OPTIMIZATION

We begin by considering the problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad a \leqq x \leqq b \end{aligned} \tag{1}$$

where f is a real function of the real variable x, and a, b are given bounds on x. We consider this problem first because it is one of the simplest optimization problems and because many of the methods for solving more complex problems require the solution of this problem. We shall describe two algorithms for solving the above problem, the first of which requires the

evaluation of the function itself only, and in the second its derivative only. (In other methods derivatives of various order may have to be computed.)

(i) Golden Section and Fibonacci Search [18-20]

The basic idea behind these two methods is to economize the number of function evaluations and trap the minimum in successively smaller nested intervals $[a_i, b_i]$. This is done by evaluating f at two interior points ℓ_i (left i) and r_i (right i) of the current interval $[a_i, b_i]$ and keeping either $[a_i, r_i]$ or $[\ell_i, b_i]$ as our new smaller interval $[a_{i+1}, b_{i+1}]$. The key to both methods is that the remaining interior point ℓ_i or r_i in $[a_{i+1}, b_{i+1}]$ is used as one of the two interior points of $[a_{i+1}, b_{i+1}]$ to further reduce it as we did for $[a_i, b_i]$. More precisely we have:

Algorithm Let $a_0 = a$, $b_0 = b$. Having a_i, b_i determine a_{i+1}, b_{i+1} as follows:

(1) Let $\ell_i = b_i - \tau_i (b_i - a_i)$

$r_i = a_i + \tau_i (b_i - a_i)$

where for

(a) Golden section: $\tau_i = \tau = (\sqrt{5} - 1)/2 \cong 0.618$

(b) Fibonacci search: $\tau_i = \dfrac{F_{n-i-1}}{F_{n-i}}$, $i=0,1,\ldots,n-3$,

$\tau_{n-2} = \dfrac{1+\varepsilon}{2}$ or $\dfrac{1-\varepsilon}{2}$, where $n \geq 2$ is a prescribed number of allowable function evaluations, ε is any small positive number

less than 1 and F_j are the Fibonacci numbers:

$$F_0 = F_1 = 1,\ F_2 = 2,\ F_3 = 3,\ F_4 = 5,\ \ldots\ F_{j+1} = F_j + F_{j-1}$$

(2) If $f(\ell_i) > f(r_i)$ take $a_{i+1} = \ell_i$, $b_{i+1} = b_i$

If $f(\ell_i) \leqq f(r_i)$ take $a_{i+1} = a_i$, $b_{i+1} = r_i$

Assumptions f is unimodal on [a,b], that is

(1) $\min_{x \in [a,b]} f(x) = f(\bar{x})$ for some $\bar{x} \in [a,b]$

(2) For ℓ, r in [a,b] and $\ell < r$

$f(\ell) > f(r)$ implies $\bar{x} \in [\ell,b]$

$f(\ell) < f(r)$ implies $\bar{x} \in [a,r]$

$f(\ell) = f(r)$ implies $\bar{x} \in [\ell,r]$

where the symbol $\in$ reads "is in".

Convergence Theorem After $n \geqq 2$ function evaluations the minimum solution $\bar{x} \in [a_{n-1}, b_{n-1}]$ where

$$b_{n-1} - a_{n-1} = (0.618)^{n-1}\ (b-a),$$

for golden section, and

$$b_{n-1} - a_{n-1} = \tau_0\tau_1\ldots\tau_{n-2}\,(b-a) = \frac{1}{F_n}\,(b-a) \text{ or } \frac{1+\varepsilon}{F_n}\,(b-a),$$

for Fibonacci search. As n approaches ∞, a_n approaches the solution $\bar{x}$ from below and b_n approaches $\bar{x}$ from above.

Remarks For a fixed number n of allowable function evaluations, Fibonacci search has the property of maximizing interval reduction. However golden section is somewhat simpler to implement, since it does not require a knowledge of n in advance. Hence in general golden section is preferred. Thus for $n = 11$ function evaluations $b_{10} - a_{10} \cong .008\ (b-a)$ for golden section and $b_{10} - a_{10} = \frac{1}{144}\ (b-a) \cong .007\ (b-a)$ for Fibonacci search, that is the interval reduction is less than 1% in both cases.

Convergence Rate If we take $b_i - a_i$ as the error in the ith step of the iteration, an estimate of the rate of convergence is given by the relation between $b_i - a_i$ and $b_{i-1} - a_{i-1}$. We have then

$$(b_i - a_i) = \tau(b_{i-1} - a_{i-1}), \quad \text{for golden section,}$$

$$(b_i - a_i) = \frac{F_{n-1}}{F_{n-i+1}}\ (b_{i-1} - a_{i-1}), \quad \text{for Fibonacci search,}$$

where $\frac{F_{n-1}}{F_{n-i+1}}$ approaches τ as n, i approach ∞. Because of the above linear relations between $(b_{i-1} - a_{i-1})$ and $(b_i - a_i)$ we say that golden section has a linear convergence rate and that Fibonacci search has an asymptotic linear convergence rate. If either the coefficient multiplying $(b_{i-1} - a_{i-1})$ above approached zero asymptotically, or if $b - a < 1$ and the exponent of $(b_{i-1} - a_{i-1})$ were bigger than one then we would have a

superlinear convergence rate which, in general, is preferrable to a linear convergence rate. We say "in general" because it may take many more function evaluations per step of the superlinear algorithm than it does in the linear algorithm in which case the faster rate per step should be reevaluated in terms of a rate per function evaluation. The next algorithm we give has indeed such a superlinear rate per function evaluation.

(ii) Secant Method [21-23]

The basic idea behind this method (also called sometimes regula falsi) is to use a discrete version of Newton's method to find a zero of the derivative $f'(x)$ of $f(x)$ in $[a,b]$ if it exists. Newton's method itself consists of linearizing $f'(x)$ around a current point and making the zero of the linearized function as the next point of the iteration. We have then the following.

Algorithm Start with distinct x_0, x_1 in $[a,b]$. Having x_i, x_{i-1} compute x_{i+1} as follows

$$x_{i+1} = x_i - \frac{f'(x_i)}{\frac{f'(x_i) - f'(x_{i-1})}{x_i - x_{i-1}}}$$

Assumptions

(1) $\left| \frac{f'''(\eta)}{2f'''(\xi)} \right| \leq M$ for all η, ξ

in some interval $[a',b']$ containing $[a,b]$.

(2) $M|\hat{x} - x_0| < 1$ and $M|\hat{x} - x_1| < 1$,

where $\hat{x}$ is a zero of f' in $[a',b']$, that is $f'(\hat{x}) = 0$.

Convergence Theorem The sequence $\{x_i\}$ converges to $\hat{x}$ such that $f'(\hat{x}) = 0$. If in addition $f''(\xi) > 0$ on $[a',b']$ then $\bar{x} = b$ if $\hat{x} > b$, $\bar{x} = \hat{x}$ if $a \leq x \leq b$, and $\bar{x} = a$ if $\hat{x} < a$, where $f(\bar{x}) = \min_{x \in [a,b]} f(x)$.

We observe here that even though certain conditions are assumed about second and third derivatives f'' and f''', they are not used in the algorithm, but only f'.

Convergence Rate For $i \geq 2$ we have

$$\left(M|x_i - \hat{x}|\right) \leq \left(M|x_1 - x|\right)^{F_{i-1}} \left(M|x_0 - \hat{x}|\right)^{F_{i-2}}$$

where F_i is the ith Fibonacci number defined earlier. If we let ε be the maximum of $M|x_1 - \hat{x}|$ and $M|x_0 - \hat{x}|$, then a bound on the error at the ith iteration is given by $\varepsilon^{F_{i-1}+F_{i-2}} = \varepsilon^{F_i}$. We have then $\frac{\varepsilon^{F_{i+1}}}{\varepsilon^{F_i}} = \varepsilon^{F_{i-1}}$ which, in the limit, approaches $\varepsilon^{.724(1.618)^{i-1}}$ as i approaches infinity. We say then that the secant method has an asymptotic convergence rate of order 1.618 per step or per function evaluation, since one function evaluation is made per step. This compares with order $\sqrt{2} = 1.414$ per function evaluation for Newton's method.

We turn our attention now to unconstrained optimization problems in n-variables.

III. UNCONSTRAINED OPTIMIZATION

We consider in this section a very important class of optimization problems, the class of problems of minimizing a real valued function of n real variables with no constraints. The problem is then

$$\text{(2)} \qquad \begin{array}{l} \text{minimize} \quad f(x) \\ x \in R^n \end{array}$$

where R^n is the n-dimensional real Euclidean space which is the collection of all possible n-dimensional vectors, and f is real valued function defined on R^n. Problem (2) can be written as

$$\text{minimize} \quad f(x^1, x^2, \ldots, x^n)$$

where $x^1, x^2, \ldots, x^n$ are the n-components of x.

Superficially this problem again may look very simple. Unfortunately it is far from being so. In fact for the general problem we cannot get a solution. All we can get is some point $\bar{x}$ which satisfies some necessary optimality condition such as

$$\nabla f(\bar{x}) \triangleq \left(\frac{\partial f(\bar{x})}{\partial x^1}, \ldots\ldots, \frac{\partial f(\bar{x})}{\partial x^n} \right) = 0.$$

If in addition we assume that f is convex, that is it is valley-like or more precisely, for any two vectors x_1 and x_2

$$f((1-\lambda)x_1 + \lambda x_2) \leqq (1-\lambda)f(x_1) + \lambda f(x_2) \quad \text{for} \quad 0 \leqq \lambda \leqq 1,$$

then $\bar{x}$ is indeed the solution. In most practical problems convexity is nonexistent or impossible to check. The best strategy then is to start the algorithm at different starting points and take the lowest of the terminal points of each start. This strategy is an advisable strategy for any optimization algorithm, constrained or unconstrained, for which the assumptions guaranteeing the minimum either do not hold or hard to check.

We divide the algorithms to be described for solving (2) into three categories depending on their rate of convergence. The linearly convergent algorithms, are extremely simple to implement, but are slow to converge. The quadratically convergent algorithms converge very rapidly per iteration, but each iteration is slow, and so the overall performance is not too good unless the number of variables is small, say less than ten. The superlinearly convergent algorithms are somewhere between the other two types of algorithms as far as complexity and convergence rate per step, however in overall performance they are vastly superior and thus they should be used in most cases.

(iii) Linear Convergence: Gradient Methods [23-27]

The basic idea here is to move along a direction p which makes a reasonably acute angle with the direction of steepest descent, the direction of the negative gradient $-\nabla f$. Two crucial things have to be monitored in order to insure convergence: the "acuteness" of the angle between p and $-\nabla f$, and the step size along p. More precisely we have the following.

Gradient Algorithm Start with any x_0. Having x_i determine x_{i+1} as follows

(1) (Direction choice) Choose any direction vector p_i such that $-\nabla f_i p_i \geqq \mu_i ||\nabla f_i||^2$ and $\|p_i\| \leqq \beta_i \|\nabla f_i\|$ where $\beta_i \geqq \mu_i \geqq 0$.

Note: For two vectors a and b, ab denotes the scalar product which is the sum of the products of their components. The norm $\|a\|$ is defined as $(aa)^{\frac{1}{2}}$.

(2) (Step size choice) Determine $x_{i+1} = x_i + \lambda_i p_i$ according to either of the following two rules:

(a) λ_i is the first nonegative root of $\nabla f(x_i + \lambda_i p_i)\, p_i = 0$, (or equivalently $\lambda_i \geqq 0$ and $f(x_i + \lambda_i p_i)$ is the first local minimum of $f(x_i + \lambda p_i)$, $\lambda \geqq 0$) provided that $\frac{\mu_i}{\beta_i} \geqq \nu > 0$ for some positive ν.

(b) λ_i is the largest of the numbers $\{\lambda, \frac{\lambda}{2}, \frac{\lambda}{4}, \ldots\}$ satisfying

$$f(x_i) - f(x_i + \lambda_i p_i) \geqq \frac{\lambda_i}{2\sigma} \| \nabla f(x_i) \|^2$$

where λ, σ are arbitrary but fixed positive numbers (λ is an initial step size) and $\frac{1}{2} + \varepsilon \leqq \sigma\mu_i \leqq \sigma\beta_i \leqq \sigma\beta$ where ε, β are arbitrary but fixed positive numbers (typically $\sigma = 1$, $\varepsilon = 10^{-6}$).

Remark If $p_i = -\nabla f_i$ then we can set $\beta_i = \mu_i = \sigma = \beta = 1$, $\varepsilon = 10^{-6}$ and satisfy all of the above requirements. We have then the classical Cauchy method [28] of steepest descent for (a) and Armijo's algorithm [27] for (b). Of course the above algorithm is more general than either.

Convergence Theorem If $\|\nabla f(y) - \nabla f(x)\| \leqq M\|y-x\|$ for some $M > 0$, and all y, x, then each accumulation point $\bar{x}$ of the sequence $\{x_i\}$ generated by the above algorithm is stationary, that is $\nabla f(\bar{x}) = 0$. If in addition f is convex, then $\bar{x}$ is the minimum solution of (2).

The assumption $\|\nabla f(y) - \nabla f(x)\| \leqq M\|y-x\|$ is a Lipschitz continuity requirement on ∇f and will be used repeadedly here. If the $n \times n$ Hessian matrix $\nabla^2 f$ of the second partial derivatives $\frac{\partial^2 f}{\partial x^k \partial x^j}, k, j = 1, \ldots, n$, exists, is symmetric and is bounded above in norm, then the Lipschitz constant M can be taken equal to the bound on this norm.

We give now a convergence rate for the special of the above algorithm when $p_i = -\nabla f(x_i)$.

Convergence Rate Let f have a stationary point $\bar{x}$, that is $\nabla f(\bar{x}) = 0$, let $\|\nabla f(y) - \nabla f(x)\| \leq M \|y-x\|$, $M > 0$, let $(\nabla f(y) - \nabla f(x))(y-x) \geq m \|y-x\|^2$, $m > 0$, and let

$$x_{i+1} = x_i - \lambda_i \nabla f(x_i), \quad 0 < \varepsilon_1 \leq \lambda_i \leq \frac{2m}{M^2} - \varepsilon_2, \ \varepsilon_2 > 0$$

Then $\{x_i\}$ converges to $\bar{x}$, and its rate of convergence is linear and is given by

$$\|x_{i+1} - \bar{x}\| \leq (1 + (\lambda_i M)^2 - 2\lambda_i m)^{\frac{1}{2}} \|x_i - \bar{x}\|.$$

If f is convex and twice continuously differentiable, M can be taken as an upper bound on the largest eigenvalue of $\nabla^2 f$ and m can be taken as a lower bound on the smallest eigenvalue of $\nabla^2 f$. We can easily check that if we pick $\lambda_i = \frac{m}{M^2}$ then we get the fastest convergence rate which is given by

$$\|x_{i+1} - \bar{x}\| \leq (1 - \frac{m^2}{M^2}) \|x_i - \bar{x}\|$$

This immediately verifies the following computationally observed fact: If $\frac{m}{M}$ is close to 1, the contours of constant f are nearly circular and we obtain fast convergence. If $\frac{m}{M}$ is close to 0, the contours of constant f are narrow ellipses and this causes slow convergence. To overcome such possible slow convergence we turn to the next class of unconstrained optimization.

(iv) Quadratic Convergence: Newton Methods [23,29]

The basic idea here can be interpreted in either of two equivalent ways: (a) Expand f in a Taylor series around the current point, keeping terms up to quadratic ones. Minimize this quadratic expansion to obtain the next point. (b) Linearize ∇f around the current point. The zero of this linearization is the next point.

We shall see that the "error" gets squared at each iteration. This implies fast convergence per iteration if the error is less than one to start with. Hence we cannot be too far off in our starting point if the error is to be less than one. (This statement however does not apply to the damped Newton algorithm below.) Also, each step is rather costly since we have to invert an $n \times n$ matrix.

Newton Algorithm Start with x_0. Having x_i compute x_{i+1} as follows

$$x_{i+1} = x_i - (\nabla^2 f(x_i))^{-1} \nabla f(x_i)$$ where $(\ )^{-1}$ denotes the matrix inverse.

Assumptions

(a) There exists an $\bar{x}$ such that $\nabla f(\bar{x}) = 0$

(b) $\|\nabla^2 f(y) - \nabla^2 f(x)\| \leq R\|y-x\|$, for some $R > 0$ and all y,x

(c) $\|(\nabla^2 f(x))^{-1}\| \leq \frac{1}{m}$, for some $m > 0$ and all x

(d) $\|x_0 - \bar{x}\| < \frac{2m}{R}$, where x_0 is the initial point

where in (b) the norm of an $n \times n$ matrix A is defined as

$\|A\| = \max \|Ax\|$ subject to $\|x\| = 1$.

(e) $m\|y\|^2 \leq y\nabla^2 f(x)y \leq M\|y\|^2$ for some M and all x,y, then the sequence $\{x_i\}$ generated by the damped Newton algorithm converges to the unique solution $\bar{x}$ of (2) at the quadratic rate

$$\|x_{i+1} - \bar{x}\| \leq \frac{M^{\frac{1}{2}}R}{2m^{\frac{3}{2}}} \|x_i - \bar{x}\|^2$$

starting with <u>any</u> x_0.

We remark that the damped Newton method uses the same direction as that as of the Newton method except that we minimize f along that direction. This is done in step (2) above. We also note that assumption (e) makes f strictly convex, and hence the minimum $\bar{x}$ is unique.

(v) <u>Superlinear Linear Convergence: Variable Metric and Conjugate Directions</u> [9-14,16,23-24,30-37]

The methods to be described in this section are among the most powerful methods for unconstrained optimization and were developed mostly in the last decade. The common idea to both variable metric and conjugate directions is that near the minimum the second-order terms of the Taylor series expansion dominate and hence, in order to have a rapidly convergent method for a general function, the method must be very fast when applied to a quadratic function. Both types of methods in fact will minimize a positive definite gradratic function in n (n is the dimensionality of the space) or less steps.

The variable metric achieves this by picking directions p_i, to minimize f along, which are close in a certain sense to the Newton direction $-(\nabla^2 f(x_i))^{-1}\nabla f(x_i)$. In fact for a quadratic function the variable metric direction at the nth step becomes the Newton direction and leads to the minimum of the quadratic function at that step. The conjugate directions methods on the other hand pick directions which satisfy the relation $p_i \nabla^2 f(x_i) p_{i-1} = 0$ where $\nabla^2 f$ is the $n \times n$ Hessian matrix. For a positive-definite quadratic function, the n first such directions are linearly independent, and by sequentially searching along these directions, the minimum of the quadratic function is arrived at the nth step.

Variable Metric Algorithm (Davidon-Fletcher-Powell) Start with any x_0. Let $H_0 = I$, the $n \times n$ identity matrix. Having x_i and H_i determine x_{i+1} and H_{i+1} as follows

(1) Let $p_i = -H_i \nabla f(x_i)$. Compute $x_{i+1} = x_i + \lambda_i p_i$ where λ_i is the first nonnegative root of $\nabla f(x_i + \lambda_i p_i) p_i = 0$, or equivalently $\lambda_i \geq 0$ and $f(x_i + \lambda_i p_i)$ is the first local minimum of $f(x_i + \lambda p_i)$, $\lambda \geq 0$.

(2) $$H_{i+1} = H_i + \frac{z_i z_i'}{z_i y_i} - \frac{(H_i y_i)(H_i y_i)'}{y_i H_i y_i}$$

where $z_i = x_{i+1} - x_i$, $y_i = \nabla f(x_{i+1}) - \nabla f(x_i)$ and $z_i z_i'$ is the

$n \times n$ matrix formed by postmultiplying the column vector z_i by the row vector z_i, and similarly for $(H_i y_i)(H_i y_i)'$.

Quadratic Case Convergence (Fletcher-Powell) If $f(x) = \frac{1}{2} x\, Cx + ax$ and C is positive definite, then the variable metric algorithms converges in n or less steps to the minimum.

General Case Convergence (Powell) If f is twice continuously differentiable and

$$m \| y \|^2 \leqq y \nabla^2 f(x) y$$

for some $m > 0$ and all x, y, then the sequence $\{x_i\}$ generated by the variable metric method converges to $\bar{x}$, the unique solution of (2). If in addition

$$\| \nabla^2 f(y) - \nabla^2 f(x) \| \leqq R \| y - x \| \quad \text{for sum} \quad R > 0 \quad \text{and all} \quad x, y$$

then

$$\| x_{i+1} - \bar{x} \| \leqq \varepsilon_i \| x_i - \bar{x} \|, \quad \lim_{i \to \infty} \varepsilon_i \to 0,$$

that is $\{x_i\}$ converges superlinearly to $\bar{x}$.

Conjugate Directions Algorithm Start with any x_0 and set $p_0 = - \nabla f(x_0)$. Having x_i, p_i determine x_{i+1}, p_{i+1} as follows

(1) $x_{i+1} = x_i + \lambda_i p_i$ where λ_i is the first nonnegative root of $\nabla f(x_i + \lambda_i p_i) p_i = 0$ or equivalently $\lambda_i \geqq 0$ and $f(x_i + \lambda_i p_i)$ is the first ideal minimum of $f(x_i + \lambda p_i)$, $\lambda \geqq 0$.

(2) $p_{i+1} = -\nabla f(x_{i+1}) + \alpha_{i+1} p_i$, where

$$\alpha_{i+1} = \frac{\|\nabla f(x_{i+1})\|^2}{\|\nabla f(x_i)\|^2} \qquad \text{(Fletecher-Reeves)}$$

or

$$\alpha_{i+1} = \frac{(\nabla f(x_{i+1}) - \nabla f(x_i))(\nabla f(x_{i+1}))}{\|\nabla f(x_i)\|^2} \qquad \text{(Polyak-Polak-Ribière)}$$

Assumption $\|\nabla f(y) - \nabla f(x)\| \leq M \|y-x\|$ for some $M > 0$ and all y,x.

Quadratic Case Convergence If $f(x) = \frac{1}{2} x c x + a x$ and C is positive definite, then the conjugate direction algorithm converges in n or less steps to the minimum.

General Case Convergence (a) Let the set $\{x \mid f(x) \leq f(x_0)\}$ be bounded, then at least one accumulation point $\bar{x}$ of the sequence $\{x_i\}$ generated by the Fletcher-Reeves conjugate directions algorithm is stationary, that is $\nabla f(\bar{x}) = 0$. (b) Let f be twice continuously differentiable, then each accumulation point $\bar{x}$ of the sequence $\{x_i\}$ generated by the Polyak-Polak-Ribière conjugate directions algorithm is stationary, that is $\nabla f(\bar{x}) = 0$.

Convergence Rate (McCormick-Ritter) (b) Let the set $\{x \mid f(x) \leq f(x_0)\}$ be bounded and let

$$m\|y\|^2 \leq y \nabla^2 f(x) y \leq M\|y\|^2 \quad \text{for some } m > 0 \text{ and all } y,x,$$

then the Polyak-Polak-Ribière conjugate directions algorithm is n-step superlinearly convergent, that is,

$$\|x_{i+n} - \bar{x}\| \leq \varepsilon_i \|x_i - \bar{x}\|, \quad \lim_{i \to \infty} \varepsilon_i = 0.$$

Remark It has been computationally observed that by resetting the conjugate direction p_i to $-\nabla f(x_i)$ every n steps improves the algorithm considerably.

IV. CONSTRAINED OPTIMIZATION

The problem here is to find $\bar{x}$ such that

$$f(\bar{x}) = \min_{x \in X} f(x), \quad \bar{x} \in X \tag{3}$$

where X is some given set in R^n. Quite often

$$X = \{x | g_j(x) \leq 0, \ j = 1, \ldots, k\} = \{x | g(x) \leq 0\}$$

where g_i are given functions, and $g = (g_1\ g_2 \ldots g_k)$. The set X is called the feasible region, the function f is called the objective function and the functions g_i are called the constraint functions or simply the constraints. If the constraints are linear the problem is considerably easier to solve than if they were nonlinear. If both the objective function and the constraints are linear, then we have a linear programming problem, the solution of which can be obtained by the standard or revised simplex algorithm [2,3]. We shall not discuss the simplex algorithm here but will use it as a subroutine in some of the algorithms given below.

It is not easy to classify the algorithms for constrained optimization as we did for the unconstrained case mainly because there are very few convergence rates available for constrained algorithms. In the absence of such rates one must rely on computational experience. Unfortunately there are very few comparative studies of constrained algorithms, the only extensive one being that of Coleville [15], which is somewhat dated, since it came to a standstill in June 1968. In the absence of conclusive theoretical or computational evidence to the superiority of some specific algorithm over others we shall give four different types of algorithms below which can be used on a wide variety of constrained problems. Algorithm (vi) is recommended for nonlinear programs of moderate size ($m + n \leqq 100$). Algorithm (vii) is recommended nonlinear programs of larger size ($m + n \geqq 100$). Algorithm (viii) is recommended for nonlinear programs with linear constraints. Algorithm (ix) is recommended for problems with a large number of convex constraints.

(vi) Penalty Function Algorithms [11,13]

The basic idea here is to reduce the constrained problem to a sequence of "unconstrained" problems the solutions of which converge to the solution of the original problem. This is done by lumping the constraints with the objective function in such a fashion that minimizing the lumped function penalizes any violation of the constraints. We shall discuss two types of penalty functions: interior and exterior ones depending on whether or not the solution

of the "unconstrained" problem is interior to or exterior to the original feasible region.

Interior Parametric Penalty Algorithm For any decreasing sequence of real positive numbers $\{r_i\}$ converging to zero find the solutions x_i of the "unconstrained" problems

$$\text{minimize} \quad P_i(x)$$

$$\text{subject to} \quad g(x) < 0$$

where $P_i(x)$ is defined below.

Remark The penalty function $P_i(x)$ is constructed such that the above problem has a solution x_i satisfying $g(x_i) < 0$. Thus x_i in essence is in the interior of the feasibile region $X = \{x \mid g(x) \leqq 0\}$ and hence any of the algorithms of unconstrained optimization can be used to solve this problem starting with an initial guess of x_{i-1}, the solution from the previous step.

Assumptions

(1) There exists some x such that $g(x) < 0$, that is the set $X^0 = \{x \mid g(x) < 0\}$ is not empty.

(2) $\underset{x \in X}{\text{infimum}} \; f(x) = \underset{x \in X^0}{\text{infimum}} > -\infty$

(3) f is lower semicontinuous

Interior Parametric Penalty Function $P_i(x) = p(f(x)) + r_i \beta(x)$, where p is a function of one variable and β a function of n variables, satisfying the following conditions:

(1) $P_i(x) = +\infty$ on the boundary of X

(2) If $\{x_i\}$ converges to x in X, then $\underline{\lim}\ \beta(x_i) \geqq \alpha(x) > -\infty$

(3) $p(f(y)) > p(f(x))$ for $f(y) > f(x)$.

Examples of Interior Parametric Penalty Functions

$$P_i(x) = f(x) - r_i \sum_{j=1}^{k} \frac{1}{g_j(x)}$$

$$P_i(x) = f(x) - r_i \sum_{j=1}^{k} \log(-g_j(x))$$

Convergence Theorem Each accumulation point $\bar{x}$ in X of the sequence $\{x_i\}$ generated by an interior parametric penalty algorithm solves the original problem (3).

In order to be able to handle cases where there is no x such that $g(x) < 0$, which is the case if we have an equality constraint $h(x) = 0$ that is replaced by two inequalities $h(x) \leqq 0$ and $-h(x) \leqq 0$, we discuss next the following exterior penalty function algorithm.

Exterior Parametric Penalty Algorithm For any increasing sequence of real positive numbers $\{r_i\}$ diverging to $+\infty$ find the solutions x_i of the unconstrained problems

$$\underset{x \in R^n}{\text{minimize}} \; P_i(x)$$

where $P_i(x)$ is defined below.

Assumptions f is continuous and the minimum of $f(x)$ on X exists.

Exterior Parametric Penalty Function $P_i(x) = p(f(x)) + r_i \beta(x)$ where p is a function of one variable and β is a function of n variables satisfying the following conditions:

(1) β is continuous on R^n and $\beta(x) = 0$ for $x \in X$, else $\beta(x) > 0$.

(2) p is continuous and increasing, that is $p(f(y)) > p(f(x))$ for $f(y) > f(x)$.

Examples of Exterior Parametric Penalty Functions

$$P_i(x) = f(x) + r_i \sum_{j=1}^{k} (g_j(x) + |g_j(x)|)^2$$

$$P_i(x) = f(x) + r_i \sum_{j=1}^{k} [g_j(x)]_+^2$$

where $[g_j(x)]_+ = \begin{cases} 0 & \text{if } g_j(x) \leq 0 \\ g_j(x) & \text{if } g_j(x) > 0 \end{cases}$

Convergence Theorem Each accumulation point $\bar{x}$ of the sequence $\{x_i\}$ generated by an exterior parametric penalty algorithm solves the original problem (3).

(vii) Feasible Directions Algorithms [38-39]

The basic idea in feasible direction methods is to find a direction which simultaneously decreases the objective function and at the same time remain feasible and then to minimize f along this direction. Unfortunately this kind of procedure without certain precautions may converge to a nonoptimal point. For example consider the problem of minimizing x, subject to $x^2 + y^2 \leq 1$. The solution is $(\bar{x}, \bar{y}) = (-1, 0)$. If we start at $(1, 0)$ and choose feasible directions by joining the current point to the halfpoint of the arc of the circle $x^2 + y^2 = 1$ joining the current point and $(0, 1)$ we will converge to $(0, 1)$ which is not optimal. To avoid such difficulties there are two procedures. One procedure uses the device of ε-active constraints [38] that is at each current point x_i of the iteration not only the active constraints $g_j(x_i) = 0$ are considered, but also the ε-active constraints also, that is $-\varepsilon \leq g_j(x) \leq 0$ for some $\varepsilon > 0$. The other device [39] is to consider all the constraints.

ε-Active Constraints Feasible Directions [38] Start with any x_0 in X and any $\varepsilon_0 = \varepsilon > 0$. Having x_i determine x_{i+1} as follows:

(1) Find a feasible direction p_i from the solution (δ_i, p_i) of the linear program

$$\underset{\delta,p}{\text{minimize}} \ \{\delta \mid \nabla f(x_i)p \leqq \delta, \ \nabla g_j(x_i)p \leqq \delta, \ j \in J(\varepsilon_i), -e \leqq p \leqq e\}$$

where e is an n-vector of ones,

$$J(\varepsilon_i) = \{j \mid -\varepsilon_i \leqq g_j(x_i) \leqq 0\}$$

and

$$\varepsilon_i = \begin{cases} \varepsilon_{i-1} & \text{if } \delta_{i-1} < -\varepsilon_{i-1} \\ \frac{1}{2}\varepsilon_{i-1} & \text{if } \delta_{i-1} \geqq -\varepsilon_{i-1} \end{cases}$$

(2) Compute $x_{i+1} = x_i + \lambda_i p_i$ in X such that

$$f(x_i + \lambda_i p_i) = \underset{x_i+\lambda p_i \in X}{\text{minimum}} \ f(x_i + \lambda p_i)$$

All-Constraints Feasible Directions [39] Start with any x_0 in X. Having x_i determine x_{i+1} as follows:

(1) Find a feasible direction p_i from the solution (δ_i, p_i) of the linear program

$$\underset{\delta,p}{\text{minimize}} \ \{\delta \mid \nabla f(x_i)p \leqq \delta, \ g_j(x_i) + \nabla g_j(x_i)p \leqq \delta$$

$$j = 1, \ldots, k, \ -e \leqq p \leqq e\}$$

where e is an n-vector of ones.

(2) Compute $x_{i+1} = x_i + \lambda_i p_i$ in X such that

$$f(x_i + \lambda_i p_i) = \underset{x_i + \lambda p_i \in X}{\text{minimum}} \quad f(x_i + \lambda p_i).$$

Assumptions f and g are continuously differentiable.

Convergence Theorem Each accumulation point $\bar{x}$ of the sequence $\{x_i\}$ generated by either of the above feasible directions algorithm is stationary, that is there exist no feasible direction p at $\bar{x}$ satisfying.

$$\nabla f(\bar{x})p < 0, \quad \nabla g_j(\bar{x})\, p < 0, \quad j \in \{j \mid g_j(\bar{x}) = 0\}$$

If in addition f and g are convex and there exist some $\hat{x}$ such that $g(\hat{x}) < 0$, then $\bar{x}$ solves the original problem.

We remark that in each iteration of the feasible direction algorithm a linear program has to be solved. This can be done by using any one of the available efficient codes for solving linear programs.

(viii) Gradient Projection Algorithms [26,40-43]

The basic idea in these methods is to try to move as close as possible to the direction of steepest descent $-\nabla f$ but still remain feasible. This is done by projecting a steepest descent step $x_i - \nabla f(x_i)$ on the feasible region X, that is find the closest point in X to $x_i - \nabla f(x_i)$.

The original and widely used gradient projection method was invented by Rosen [40] and with second order modification by Goldfarb [42] has become one

of the fastest algorithms for linearly constrained problems [43]. This algorithm is a little complex to describe in a short survey as this, so we shall describe the closely related and more recent Levitin-Polyak gradient projection algorithm [26]. Although the Levitin-Polyak algorithm is somewhat more difficult to implement (it requires for the general case the solution of a quadratic program at each iteration) it can give explicit simple iterations for equality constraints alone or bound inequalities alone. Also Rosen's gradient projection algorithm can be derived from the Levitin-Polyak algorithm.

Levitin-Polyak Gradient Projection Start with any x_0 in X. Having x_i determine x_{i+1} as follows

(1) Find the projection y_i of $x_i - \nabla f(x_i)$ on X, that is solve the quadratic programming problem

$$\underset{y \in X}{\text{minimize}} \; \| y - (x_i - \nabla f(x_i)) \|^2$$

and call its solution y_i.

(2) Let $p_i = y_i - x_i$. Then $x_{i+1} = x_i + \lambda_i p_i$ where λ_i is determined by either of the following two methods:

(a) $f(x_i + \lambda_i p_i) = \underset{0 \leq \lambda \leq 1}{\text{minimum}} \; f(x_i + \lambda p_i)$

(b) λ_i is the largest of the numbers $\{\lambda, \frac{\lambda}{2}, \frac{\lambda}{4}, \ldots\}$ satisfying

$$f(x_i) - f(x_i + \lambda_i p_i) \geq \frac{\lambda_i}{2} \| p_i \|^2$$

where $\lambda > 0$ is an arbitrary but fixed positive number.

Assumptions

(1) X is convex and closed(A sufficient condition for this is that g is convex and continuous)

(2) $\| \nabla f(y) - \nabla f(x) \| \leq M \| y - x \|$, for some $M > 0$ and all y,x.

Convergence Theorem Each accumulation point $\bar{x}$ of the sequence $\{x_i\}$ generated by the gradient projection algorithm satisfies the necessary optimality condition

$$\underset{y \in X}{\text{minimum}} \| y - (\bar{x} - \nu \nabla f(\bar{x})) \|^2 = 0 \quad \text{for some} \quad \nu > 0.$$

If in addition f is convex, then $\bar{x}$ is a solution of the original problem (3).

We note that the bulk of the work of this algorithm has in solving the quadratic program of step (1). We note that if the constraints are linear then this quadratic program can be solved in a finite number of iterations by using any of the algorithms described in [44-45]. For some special cases, the solution of this quadratic program can be written in one step. We give two such special cases below.

Special Case: Equality Constraints: If the feasible region is $X = \{x | Ax = b\}$ where A is $k \times n$ matrix of rank k then the solution y_i of the quadratic program of step (1) is given by

$$y_i = z_i - A' (AA')^{-1} (Az_i - b)$$

where $z_i = x_i - \nabla f(x_i)$, and the prime denotes the transpose.

Special Case: Bounded Variables If the feasible region is $X = \{x | a \leqq x \leqq b\}$, where a and b are given n-vectors then the solution y_i of the quadratic program of step (1) is given as follows

$$(y_i)^j = (x_i - \nabla f(x_i))^j \quad \text{if} \quad a^j < (x_i - \nabla f(x_i))^j < b^j$$

$$(y_i)^j = a^j \quad \text{if} \quad (x_i - \nabla f(x_i))^j \leqq a^j$$

$$(y_i)^j = b^j \quad \text{if} \quad b^j \leqq (x_i - \nabla f(x_i))^j$$

where the j superscript denotes the jth component of the vector.

(ix) Cutting Plane Algorithms Without Nesting [46-50]

The basic idea of cutting plane algorithms is to replace the given optimization problem on the set X by optimization problems on a sequences $\{X_i\}$ of sets containing X and more simple than X itself. In the original cutting plane methos [46-47] each X_i was contained in X_{i-1} and was obtained from X_{i-1} by cutting part of X_{i-1} with a cutting plane. (This in essence is the Remez first algorithm of Chebyshev approximation [50].) In the more recent algorithms [48-49] the nesting requirement, that is X_i is in X_{i-1}, which can cause difficulties such as large number of constraints and linear dependence of the constraints, has been eliminated. (This newer type of algorithm is the Remez second algorithm for Chebyshev approximation [50].)

Cutting Plane Algorithm Start with ℓ_0 points $t_1, t_2, \ldots, t_{\ell_0}$ in R^n. At iteration i we have ℓ_i points $t_1, t_2, \ldots, t_{\ell_i}$. We determine x_i and ℓ_{i+1} points in R^n as follows:

(1) Compute x_i by solving the linearized problem

$$\underset{x}{\text{minimize}} \quad f(x)$$

$$\text{subject to} \quad g_{r(t_j)}(t_j) + \nabla g_{r(t_j)}(t_j)(x - t_j) \leqq 0, \; j = 1,2,\ldots,\ell_i$$

where $g_{r(t_j)}(t_j) = \text{maximum } \{g_1(t_j), g_2(t_j), \ldots, g_k(t_j)\}$

and if there is more than one $g_{r(t_j)}$, take any one.

(2) The ℓ_{i+1} points consist of the ℓ_i points, less all those t_j for which the dual variable of constraint j were zero in step (1), plus x_i, the solution of step (1). (It can be shown that $\ell_{i+1} \leqq n + 1$.)

Assumptions

(1) f is continuous and convex

(2) g is convex and has continuous first partial derivatives

(3) For each i and any ℓ_i points the solution x_i of step (1) exists, is unique and is contained in some fixed bounded set.

Convergence Theorem Each accumulation point $\bar{x}$ of the sequence $\{x_i\}$, and there exists at least one such accumulation point, solves the original problem (3).

Remarks In order to make the problem of step (1) a linear programming problem the original programming problem can be replaced by the equivalent problem in the $n + 1$ variables x, ζ:

$$\begin{aligned} &\underset{x,\zeta}{\text{minimize}} && \zeta \\ &\text{subject to} && f(x) - \zeta \leqq 0 \\ & && g(x) \leqq 0. \end{aligned}$$

The uniqueness in assumption (3) above is the most restrictive assumption which seems to be necessary in cutting plane algorithms without nesting.

BIBLIOGRAPHY

1. F. John: "Extremum problems with inequalities as subsidiary conditions," Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948 (Courant Anniversary Volume), Interscience, New York, 1948, 187-204.

2. G. B. Dantzig: "Linear Programming and Extensions," Princeton University Press, Princeton, 1963.

3. S. I. Zukhovitskiy & L. I. Avdeyeva: "Linear and Convex Programming," (translated from Russian) Saunders, Philadelphia, 1966.

4. C. Berge & A. Ghouila-Houri: "Programming, Games & Transporation Networks," (translated from French) Wiley, New York, 1965.

5. O. L. Mangasarian: "Nonlinear Programming," McGraw-Hill, New York, 1969.

6. J. Stoer & C. Witzgall: "Convexity and Optimization in Finite Dimensions I," Springer, New York, 1970.

7. R. T. Rockafellar: "Convex Analysis," Princeton University Press, Princeton, 1970.

8. H. P. Künzi & W. Oettli: "Nichtlineare Optimierung: Neuere Verfahren Bibliographie," Lecture Notes in Operations Research and Mathematical Systems No. 16, Springer, Berlin, 1969.

9. J. Abadie (editor): "Integer and Nonlinear Programming," North-Holland, Amsterdam, 1970.

10. R. Fletcher (editor): "Optimization," Academic Press, London, 1969.

11. A. V. Fiacco & G. P. McCormick" "Nonlinear Programming: Sequential Unconstrained Minimization Techniques," Wiley, New York, 1968.

12. J. B. Rosen, O. L. Mangasarian & K. Ritter (editors): "Nonlinear Programming," Academic Press, New York, 1970.

13. W. I. Zangwill: "Nonlinear Programming: A Unified Approach," Prentice-Hall, Englewood Cliffs, N. J., 1969.

14. E. Polak: "Computational Methods in Optimization: A Unified Approach," Academic Press, New York, 1971.

15. A. R. Colville: "A Comparative Study on Nonlinear Programming Codes," IBM Philadelphia Scientific Center, Report No. 320-2949, June 1968.

16. A. M. Vercoustre: "Etude comparative des méthodes de minimisation de Fletcher et Powell et de Davidon," Electricite de France, Bulletin de la Direction des Etudes et Recherches, Serie C, No. 1, 1970, 57--76.

17. P. Wolfe: "Convergence Theory in Nonlinear Programming," in [9], 1-36.

18. J. Kowalik & M. R. Osborne: "Methods for Unconstrained Optimization Problems," Elsevier, New York, 1968.

19. D. J. Wilde: "Optimum Seeking Methods," Prentice-Hall, Englewood Cliffs, N. J., 1964.

20. K. Overholt: "An instability in the Fibonacci and golden section search methods," BIT 5, 1965, 284-286.

21. A. M. Ostrowski: "Solution of Equations and Systems of Equations," Second Edition, Academic Press, New York, 1966.

22. E. Isaacson & H. B. Keller: "Analysis of Numerical Methods," Wiley, New York, 1966.

23. J. M. Ortega & W. C. Rheinboldt: "Iterative Solution of Nonlinear Equations in Several Variables," Academic Press, New York, 1970.

24. J. W. Daniel: "The Approximate Minimization of Functionals," Prentice-Hall, Englewood Cliffs, N. J., 1971.

25. A. A. Goldstein: "Constructive Real Analysis," Harper & Row, N. Y., 1967.

26. E. S. Levitin & B. T. Polyak: "Constrained minimization methods," USSR Computational Mathematics and Mathematical Physics, 6 (5), 1966, 1-50.

27. L. Armijo: "Minimization of functions having Lipschitz continuous first partial derivatives," Pacific J. Math. 16, 1966, 1-3.

28. A. Cauchy: "Methode generale pour la resolution des systemes d'equations simultanees," C. R. Acad. Sci. Paris 25, 1847, 536-538.

29. R. A. Tapia: "The Kantorovich theorem for Newton's method," Amer. Math. Monthly. 78 (4), 1971, 389-392.

30. W. C. Davidon: "Variable metric method for minimization," Atomic Energy Commission Report ANL-5990, Argonne National Laboratory, Lemont, Ill., 1959.

31. R. Fletcher & M. J. D. Powell: "A rapidly convergent descent method for minimization," Computer J. 6, 1963, 163-168.

32. M. J. D. Powell: "On the convergence of the variable metric algorithm," Atomic Energy Research Establishment, Harwell, T. P. 382, October 1969.

33. R. Fletcher & C. M. Reeves: "Function minimization by conjugate gradients," Computer J. 7, 1964, 149-154.

34. B. T. Polyak: "The method of conjugate gradients in extremum problems," USSR Computational Mathematics & Mathematical Physics, 9 (4), 1969, 807-821 (Russian)

35. E. Polak & G. Ribiere: "Note sur la convergence de methodes de directions conjugees, Rev. Fr. Inform. Rech. Operation. 16-R1, 1969, 35-43.

36. G. Zoutendijk: "Nonlinear programming, computational methods," in [9], 37-86.

37. G. P. McCormick & K. Ritter: "On the convergence and rate of convergence of the conjugate gradient method," MRC Technical Summary Report #1118, December 1970, University of Wisconsin, Madison.

38. G. Zoutendijk: "Methods of Feasible Directions," Elsevier, Amsterdam, 1960.

39. D. M. Topkis & A. Veinott, Jr.: "On the convergence of some feasible directions algorithms for nonlinear programming," J. SIAM Contr. 5 (2), 1967, 268-279.

40. J. B. Rosen: "The gradient projection method for nonlinear programming," I, II SIAM J. Appl. Math. 8, 1960, 180-217 and 9, 1961, 514-532.

41. R. L. Fox: "Optimization Methods for Engineering Design," Addison-Wesley, Reading, Mass., 1971

42. D. Goldfarb: "Extension of Davidon's variable metric method to maximization under linear inequality constraints," SIAM J. Appl. Math. 17 (4), 1969, 739-764.

43. J. Kreuser: "GPM/GPMNLC Extended Gradient Projection Method - Nonlinear Programming Routine" University of Wisconsin Computing Center, Madison, June 1971.

44. P. Wolfe: "The simplex method for quadratic programming," Econometrica, 27 (3), 1959, 382-399.

45. R. W. Cottle & G. B. Dantzig: "Complementary pivot theory of mathematical programming," Linear Algebra and its Applications 1, 1968, 103-125.

46. E. W. Cheney & A. A. Goldstein: "Newton's method for convex programming and Tchebycheff approximation," Numer. Math. 1, 1959, 253-268.

47. J. E. Kelley: "An application of linear programming to curve fitting," SIAM J. Appl. Math. 6, 1958, 15-22.

48. D. M. Topkis: "A note on cutting-plane methods without nested constraint sets," Operations Research, 18(6), 1970, 1216-1220.

49. K. R. Gehner: "Optimization with infinite number of constraints and applications to approximation theory," Ph.D. Thesis, Computer Sciences Department, University of Wisconsin, Madison, July 1971.

50. E. W. Cheney: "Introduction to Approximation Theory," McGraw-Hill, New York, 1966.