MANAGEMENT SCIENCE
Vol. 22, No. 1, September 1975
Printed in U.S.A.

AN EFFICIENT ONE DIMENSIONAL SEARCH PROCEDURE*

R. L. FOX,† L. S. LASDON,§ ARIE TAMIR¶ AND MARGERY RATNER‡

Many nonlinear programming algorithms utilize a one-dimensional search along directions generated by the algorithm. This paper describes a method for performing this search. The method finds 3 points which bracket the minimum, fits a quadratic through them to yield a fourth point, then fits successive cubics through 4 points, discarding one each time, until certain stop criteria are met. No gradient evaluations are required. Detailed flow charts of this procedure are given, and its performance is compared with that of 2 other algorithms. Eight test problems are used in this comparison, each solved using both exterior and interior penalty functions. The Davidon-Fletcher-Powell method is used to generate the search directions. Results show that the proposed procedure requires about ½ to ¾ the computer time of its nearest competitor, a procedure designed to be especially efficient when applied to penalty functions, and about ⅓ the time of the other competitor, the 2 point cubic search using derivatives.

1. Introduction

Most efficient methods for unconstrained minimization utilize a one-dimensional search along directions generated by the method. If P is the function to be minimized, X the current vector of decision variables, and S the search direction, then the one-dimensional search problem is to choose $\alpha > 0$ yielding the first local minimum of $P(X + \alpha S)$. A significant portion of the total computational effort is expended in this search. The problem can be particularly difficult when P is an interior penalty function. This is a situation of great practical importance because penalty functions are widely used.

The most popular one-dimensional search procedures for use in unconstrained minimization utilize quadratic [3], [7] or 2 point cubic [2], [3], [4] interpolation of $P(X + \alpha S)$. When applied to penalty functions these interpolation approaches have serious deficiencies. Quadratic interpolation has the drawback that its order of convergence is approximately 1.3, significantly less than that of 2 point cubic interpolation, which is 2 [9]. The 2-point cubic, however, requires the computation of ∇P. This is usually time consuming and is often difficult to code. In some cases ∇P may not be available analytically.

In [5] a special purpose one-dimensional search procedure for interior penalty functions is described whose performance is superior to that of quadratic and 2-point cubic interpolation. The procedure is based on substituting interpolations of the objective and constraint functions into the penalty function and then minimizing this approximation of P along the current search direction.

This paper describes a one-dimensional search based on quadratic and cubic interpolations. These are obtained using function values only. Computational results on interior and exterior penalty functions show that the method is considerably faster than any of the above mentioned techniques.

* Processed by Professor Arthur M. Geoffrion, Departmental Editor for Integer and Large-Scale Programming; received June 13, 1973, revised April 1, 1974. This paper has been with the authors 3 months for revision.

† Associate Professor of Engineering, Case Western Reserve University.
§ Professor of Operations Research, Case Western Reserve University.
¶ Assistant Professor, Graduate School of Management, Northwestern University.
‡ Programming Consultant, Computer and Information Sciences, Case Western Reserve University.

2. Description of the Algorithm

We assume initially that S is a direction of descent i.e. $S^T \nabla P(X) < 0$. This assumption is relaxed later.

The procedure starts with a search for three points, A, B, and C along the direction $X + \alpha S$ which satisfy

$$XA = X + A \cdot S, \quad XB = X + B \cdot S,$$

$$XC = X + C \cdot S, \quad 0 \leqq A < B < C,$$

where $P(XA) > P(XB)$ and $P(XC) > P(XB)$.

The program logic for doing this is diagrammed in Figure 1. In block 1 there is a requirement for an initial step size, T_0. With the Davidon Fletcher Powell (*DFP*) method [4] or other Variable Metric Methods, T_0 is equal to the optimal α value from the previous search except when the *DFP* method is started or restarted. Then T_0 is

$$T_0 = 0.1 \max | x_i | / \max \left| \frac{\partial P}{\partial x_i} \right| . \tag{2.1}$$

The theoretical basis for this is that, as a Variable Metric Method converges, the optimal α values should converge to 1, the optimal step for Newton's Method. Hence

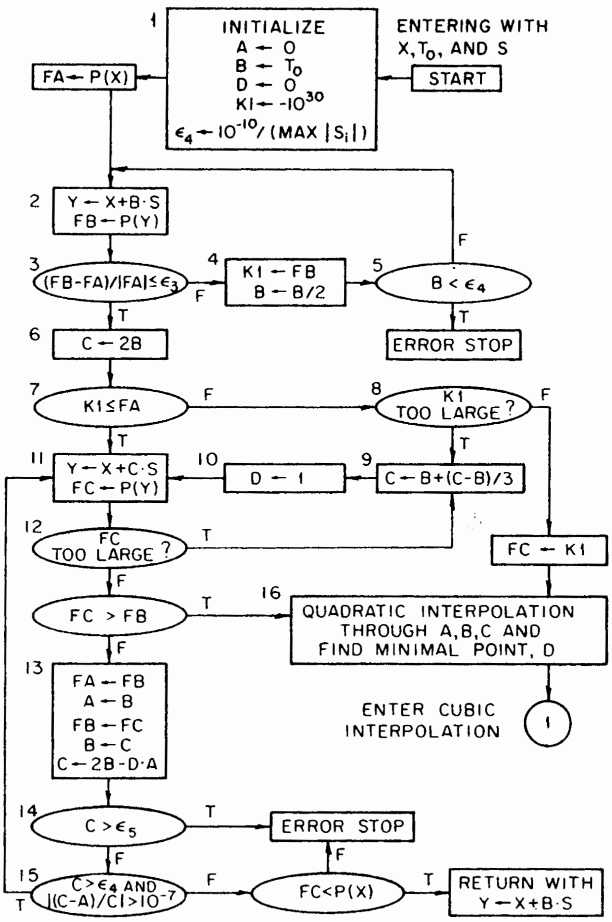

FIGURE 1. Minimum Bracketing and Quadratic Interpolation.

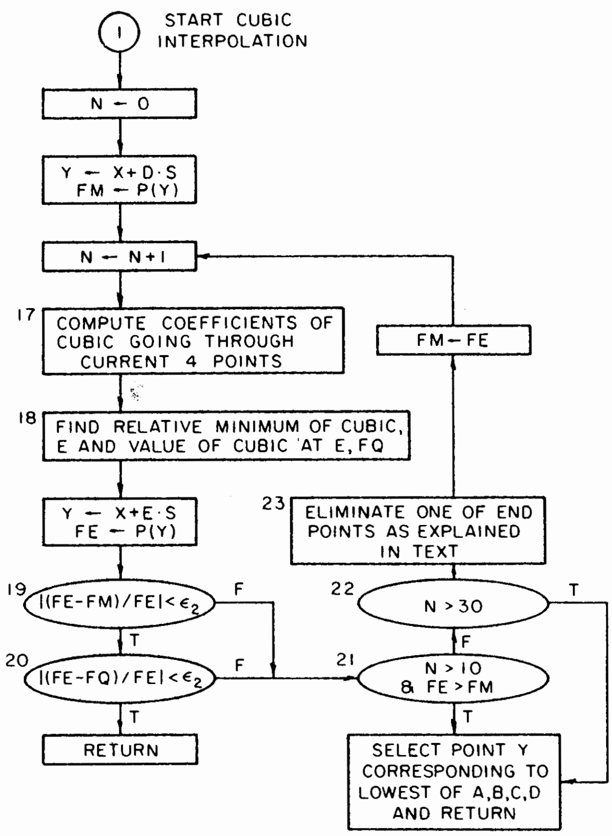

FIGURE 2. Cubic Interpolation.

the previous optimal step is a good approximation of the current one. When restarting, with S equal to $-\nabla P$, an optimal step much smaller than unity is generally taken. Relation (2.1) assumes a scaling of the variables such that a change of 0.1 max $| x_i |$ in any variable causes a small but significant change in the function being minimized.

The normal exit from the loop 2-3-4-5 is to 6 with a point B such that $FB < FA$. Block 3 also permits this exit if FB is slightly larger than FA, to allow for numerical error in evaluating P. The current value of ϵ_3 is 10^{-7}. Block 5, which uses ϵ_4 equal to $10^{-10}/(\max | s_i |)$, provides an error stop which is useful when there are errors or discontinuities in the function or its gradient. The test in block 7 is false only if the step size has been halved at least once in 2-3-4-5 in which case $K1$ is the function value corresponding to C. The test in block 8 is to prevent the situation shown in Figure 3 which is not well interpolated by a quadratic. If, in block 8, $K1$ is not "too large" we proceed directly to quadratic interpolation. The test for "too large" should be an upper bound on the ratio $(FC - FB)/| FA - FB |$ although in our working program a slightly different form was used. If $K1$ is too large then block 9 generates a new C point 1/3 of the distance from B to C. The loop 9-10-11-12 is traversed until FC is not too large.

When P is an interior penalty function a value of $P = 10^{30}$ is returned when the trial point Y is infeasible. Hence the loop 9-10-11-12 has the effect of finding a feasible point fairly rapidly.

With $D = 0$ (which occurs if and only if a $K1$ or FC which was too large has never

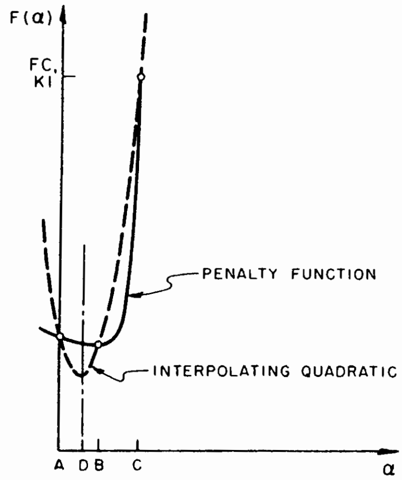

FIGURE 3. Poor Interpolation by a Quadratic.

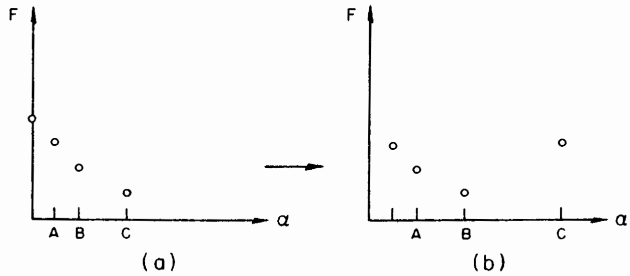

FIGURE 4. Bracketing the Minimum.

been generated) the loop 11-15 transforms the points A, B, C in Figure 4(a) into those shown in Figure 4(b). The step size is doubled each time until the points A, B, C bracket the minimum. If FC ever becomes too large D is set to 1 (block 10). Then 11-15 transforms points as shown in Figures 5(a) thru 5(c). Instead of doubling the step, a constant increment, B-A, is added.[1]

The error stop in block 14 protects against a runaway condition where P has decreased indefinitely along the search direction. Currently $\epsilon_5 = 10^{10}$.

The quadratic interpolation in block 16 yields a 4th point, D, with function value FM, somewhere between A and C. In blocks 17 and 18 a cubic polynomial is passed through the four points, FA, FB, FC, FM with its minimum at the point E. The optimality tests in 19 and 20 are passed if the percentage difference between: (a) the P values at the current and previous interpolated points and (b) the values of P and

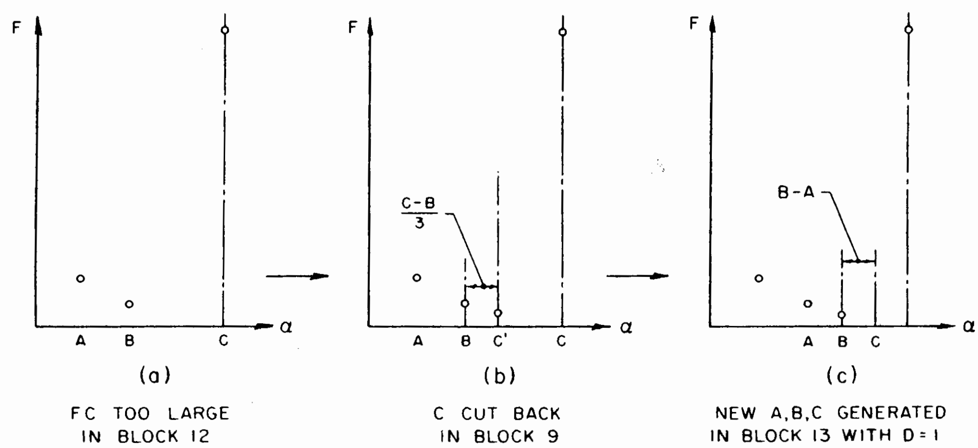

FIGURE 5. Seeking a "Good" Interpolation Set.

[1] This scheme was originally suggested by Professor K. D. Willmert of Clarkson College.

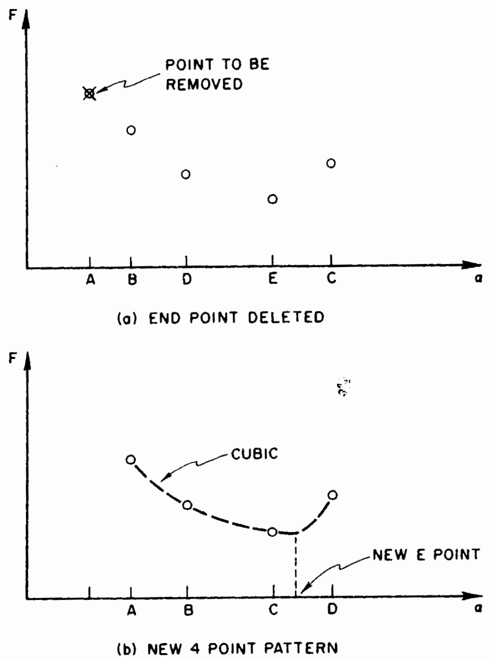

FIGURE 6. The Cubic Interpolation.

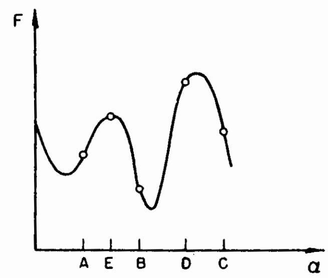

FIGURE 7. Non-Unimodality.

the cubic at E are sufficiently small. Currently $\epsilon_2 = 10^{-4}$. The tests in blocks 21 and 22 are true only rarely and prevent too many interpolations from being performed.

In block 23 the usual situation is as shown in Figures 6(a) and 6(b). Removal of an end point leaves 4 points which bracket the minimum and these are used in the next cubic fit. If a bracket cannot be formed by removal of an end point the highest end point is discarded and a new cubic fit is performed. As an example, Figure 7 shows the graph of the function being approximated. The minimum of the cubic is at E, no bracket can be formed, and point C is discarded.

3. Computational Results

Eight test problems were solved to test this one-dimensional search procedure. These are all nonlinear programs with inequality constraints. Each has the form

$$\text{minimize } f(x), \quad \text{subject to } g_i(x) \geqq 0, \quad i = 1, \ldots, m.$$

Each was solved by a sequence of minimizations of the interior penalty function

$$P(x, r) = f(x) + r \textstyle\sum_{i=1}^{m} 1/g_i(x) \tag{3.1}$$

and by a sequence of minimizations of the exterior penalty function

$$G(x, r) = f(x) + k \textstyle\sum_{i=1}^{m} (\min(0, g_i(x)))^2. \tag{3.2}$$

The first four of these are test problems 3 through 6 of [5], while the last four arise from engineering design. Problem characteristics are displayed in Figure 8 below:

Problem	1	2	3	4	5	6	7	8
No. of Vars.	4	5	9	15	8	7	8	18
No. of Constr.	3	6	13	5	29	26	516	516
No. of Bounds	2	10	1	15	6	7	8	18
Nature of Objective	Quad-ratic	Quad-ratic	Quad-ratic	Cubic	Linear	Linear	Linear	Linear
Nature of Constraints	Quad-ratic	Quad-ratic	Quad-ratic	Quad-ratic	Ratio of Polynomials	See below	See below	See below

FIGURE 8. Problem Characteristics.

Test problem 5 arises from the design of a pair of concentric helical compression springs for minimal spring stiffness. The nonlinear constraint functions were all ratios of polynomials with numerous cross product terms and exponents as high as 6. Test problems 6–8 arise from the design of statically indeterminate structures consisting of straight members connected by hinged joints (i.e. trusses). A linear stress-strain relation was used. This implies that the deflections, y, of the joints of the structure satisfy

$$[A(x)]y = b. \tag{3.3}$$

The elements of the square nonsingular matrix A are linear functions of the design vector x, whose components are the cross-sectional areas of the members, and b is a fixed load vector. The stresses in the members σ are given by $\sigma = Ry$ where R is a known matrix. The constraints are

$$0 < x_i^l \leqq x_i \quad \text{and} \tag{3.4}$$

$$y_i^l \leqq y_i \leqq y_i^u, \tag{3.5}$$

$$\sigma_j^l \leqq \sigma_j \leqq \sigma_j^u, \tag{3.6}$$

where the u and l superscripts denote given upper and lower bounds.

Since y is a nonlinear function of x through (3.1), the constraints (3.5) and (3.6) are nonlinear. The large numbers of constraints in problems 7 and 8 arose from the presence of multiple load conditions (several b vectors) each of which generates a set of σ and y vectors. Problem 7 had certain symmetries which implied that the optimal solution would have several members with equal areas. In problem 7 these areas were denoted by the same x_i variables. Problem 8 differed from 7 only in that different variables were assigned to some areas. Optimal solutions to both problems are the same.

A more complete description of problem 6 can be found in [2] and problems 7 and 8 are described in [10].

Figure 9 displays the results of solving the 8 test problems by both interior and exterior penalty methods using 3 one-dimensional search procedures. The first of

problem \ method		Exterior Penalty Function			Interior Penalty Function		
		Special Purpose	4 point Cubic	2 point Cubic	Special Purpose	4 point Cubic	2 point Cubic
		1	2	3	4	5	6
1	Function Calls	575	390	—	199	320	672
	Time (Sec)	0.9	0.7		0.8	0.6	0.7
2	Function Calls	306	406	—	682	635	1665
	Time	1.0	0.7		2.0	1.1	1.7
3	Function Calls	1851	406	—	286	347	1653
	Time	1.8	1.3		1.4	0.9	1.4
4	Function Calls	5761	2715	—	3123	2974	10969
	Time	6.2	4.8		10.1	6.4	10.4
5	Function Calls	FAILED	FAILED	FAILED	3320	3208	12151
	Time				16.7	9.5	36.0
6	Function Calls	2552	1528	3472	2461	1966	6982
	Time	46.0	28.2	60.7	46.2	36.6	113.9
7	Function Calls	3249	1795	3690	1427	993	3029
	Time	330.0	183.0	357.0	203.5	106.6	316.5
8	Function Calls	3952	1904	—	394	347	989
	Time	339.0	194.0		54.2	36.5	99.5
	Total Function Calls	18,246	9,144	—	8,572	7,582	25,959
	Total Time	725	412	—	318	189	544

FIGURE 9. Computational Results—3 One-Dimensional Search Algorithms.

these, the "special purpose" method, is designed to exploit the special structure of penalty functions. Its application to interior penalty functions is described in [5]. The adaption to exterior functions is straightforward, and is described in [6]. The second algorithm studied is the authors' version of the popular 2 point cubic interpolation procedure ([1], [5], [7]). A flow chart of this is given in [5]. The third method is the 4 point cubic search described previously.

The upper number in each box of Figure 9 gives the equivalent function evaluations—function calls plus (number of variables) $\times$ (gradient calls). The lower number is the execution time in seconds. All programs were coded in FORTRAN V and run on a Univac 1108. The results in the first three columns were obtained by sequential minimization of the exterior penalty function (3.1). This function was minimized using Goldfarb's algorithm [4]. All upper and lower bounds were incorporated directly by this algorithm, and were not included in the penalty function. The interior penalty functions in columns 4 through 6 were minimized using the *DFP* method, with the H matrix reset every $n + 1$ iterations (n = number of variables). For each test problem, except for number 8, the penalty function was minimized for several values of the penalty parameter. In problem 8, only one value of the penalty parameter was used.

It is evident that the 4 point cubic search is the best of the 3, for both interior and exterior penalty functions. On the larger problems (5 through 8) its run times and function evaluations are both 1/2 to 3/4 those of its nearest competitor, the special purpose search. The two point cubic search is far worse. This became evident while minimizing the interior penalty functions, and it was applied to only 3 exterior penalty problems. We found these results initially surprising, since the special purpose searches are designed to exploit the structure of the penalty functions. Superiority of the 4 point cubic search is due in large part to (1) the fact that it uses no derivatives; these often take much time to compute, and (2) the high "overhead" of the special purpose method—in each of 2 or more stages, it interpolates each constraint function, then minimizes an approximating function using Newton's method. The logic by which points are selected for the quadratic and cubic fits is also an important factor in the efficiency of the 4 point cubic method. The logic of Figures 1 and 2 has evolved from earlier quadratic and cubic interpolation routines.

5. Conclusion

A one-dimensional search procedure has been presented whose performance on 8 test problems involving penalty functions is significantly better than that of competing methods. The algorithm does not require derivatives, so it may be used with unconstrained minimizers such as Powell's method [9]. In this case, one must provide for the case $\alpha < 0$, since the search directions need not be downhill. This may be done by introducing a variable K_2, similar to K_1 in Figure 1, and by adding steps to blocks 2-3-4-5-6 to check $-B$ in addition to $+B$. The method incorporates safeguards to eliminate unbounded solutions and to detect directions in which no improvement can be made. It is currently used in a number of interactive and batch penalty function codes at CWRU, and all experience with it thus far has been good. Its use in conjunction with a variety of unconstrained minimizers is recommended.

References

1. Fletcher, R., and Powell, M. J. D., "A Rapidly Convergent Descent Method for Minimization," *Brit. Computer J.*, 6, 1963, pp. 163–168.
2. Fox, R. L., *Optimization Methods for Engineering Design*, Addison-Wesley Pub. Co., Reading Mass., 1971.
3. Fletcher, R. and Reeves, C. M., "Function Minimization by Conjugate Gradients," *Brit. Computer J.*, 7, 1964, pp. 149–154.
4. Goldfarb, D., "Extension of Davidon's Variable Metric Method to Maximization Under Linear Inequality and Equality Constraints," *SIAM J. Appl. Math.*, 17, No. 4, 1969 pp. 739–764.
5. Lasdon, L. S., Fox, R., and Ratner, M., "An Efficient One-Dimensional Search Procedure for Barrier Functions," to appear in *Mathematical Programming*.
6. ——, ——, and Tamir, A., "Nonlinear Programming using Exterior Penalty Functions," report prepared for ONR contract number N0014-67-A-0404-0010, Case Western Reserve University, November 1972.
7. Lootsma, F. A., "Penalty-Function Performance of Several Uncontsrained Minimization Techniques," *Phillips Res. Repts.*, 27, 1972, pp. 358–385.
8. Luenberger, D., *Introduction to Linear and Nonlinear Programming*, Addison-Wesley Publishing Co., 1973, Sections 7.2, 7.3.

9. Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Using Derivatives," *Brit. Computer J.*, 7, 1964, pp. 155–162.

10. Venkayya, V. B., Khot, N. S., and Reddy, V. S., "Optimization of Structures Based on the Study of Strain Energy Distribution," Second Conference on Matrix Methods in Structural Mechanics, Oct. 1968, AFFDL-TR-68-150.