Optimization by Direct Search and Systematic Reduction of the Size of Search Region

A direct search procedure utilizing pseudo random numbers over a region is presented to solve nonlinear programming problems. After each iteration the size of the region is reduced so that the optimum can be found as accurately as desired. The ease of programming, the speed of convergence, and the reliability of results make the procedure very attractive for solving nonlinear programming problems.

REIN LUUS and T. H. I. JAAKOLA

Department of Chemical Engineering and Applied Chemistry
University of Toronto
Toronto, Canada

SCOPE

The objective of this study was to develop a simple optimization procedure which would be applicable to solve nonlinear programming problems. We wanted to have a procedure which would be sufficiently simple to use so that anyone interested in system optimization could use it. Also, we wanted to develop a procedure which would be general enough so that it could be applied to any region of interest and to any type of functions.

The procedure presented in this paper meets our main objectives since it is extremely simple to use and is very effective in solving rather complex problems. In fact, our experience with several problems shows that the procedure could very well become as standard as linear programming. The method is straightforward and even simpler to use than linear programming since no complicated subroutine is required. A typical problem presented in this paper can be set up within a couple of hours and solved in a few seconds of computer time.

CONCLUSIONS AND SIGNIFICANCE

The optimization procedure based on direct search and systematic search region reduction is formulated and is found effective in solving various problems in the field of nonlinear programming. In this paper 6 sample problems are used to illustrate the procedure and to test its effectiveness. In every case the optimum could be reached in less than 10 seconds of computation time on an IBM 370/165 computer.

The procedure handles either inequality or equality constraints and the feasible region does not have to be convex. No approximations or auxiliary variables are required.

The most attractive features of the procedure are the ease of setting up the problem on the computer, speed in obtaining the optimum, and the reliability of the results.

In engineering design and operation one is frequently confronted with the problem of static nonlinear optimization. In design one wants to find the bottlenecks and redesign certain aspects of the system to maximize some function, for example return on investment. In operation one wants to know what flow rates, recycle, reactor temperature, catalyst activity, etc. to use to maximize some criterion of operation, such as the daily profit. Since the relationships between many variables could be nonlinear and the objective function may also be nonlinear, one confronts a nonlinear programming problem.

We may formulate the nonlinear programming problem as follows:

Maximize (or minimize)

$$P = f(x_1, x_2, \ldots, x_n) \tag{1}$$

subject to the constraints

$$g_i\ (x_1, x_2, \ldots, x_n) \leq 0 \quad i = 1, 2, \ldots, m \tag{2}$$

$$h_j\ (x_1, x_2, \ldots, x_n) \geq 0 \quad j = 1, 2, \ldots, r \tag{3}$$

$$q_k\ (x_1, x_2, \ldots, x_n) = 0 \quad k = 1, 2, \ldots, s \tag{4}$$

In order to have a meaningful problem, we must restrict the number of equality constraints s to be less than the number of variables n; there is, however, no limit to the number of inequalities. Thus $s < n$, and no restrictions are placed on m or r.

Various authors suggest sectional linearization of nonlinear functions, thus converting the nonlinear programming problem into a linear programming problem. For example, such an approach was used by Hovanessian and Stout (1963) and Hovanessian and Pipes (1969) in optimizing the fuel allocation in power plants, and by Di Bella and Stevens (1965) in optimizing a complex nonlinear reactor system. However, such approximations are not at all necessary if the following approach is considered.

DIRECT SEARCH PROCEDURE

In most of the problems we have encountered, each equality constraint can be readily removed by solving each equation in terms of a different variable seriatim. Thereby the number of variables to be determined is reduced by the number of equalities. For example, suppose we take the problem of determining x_1, x_2, and x_3 which will maximize the function P given by

$$P = f(x_1, x_2, x_3) \tag{5}$$

subject to the constraints

$$g_1(x_1, x_2, x_3) \leq 0 \tag{6}$$

$$g_2(x_1, x_2, x_3) \geq 0 \tag{7}$$

$$g_3(x_1, x_2, x_3) \geq 0 \tag{8}$$

$$g_4(x_1, x_2, x_3) = 0 \tag{9}$$

We assume that some operation can be performed on the equality constraint so that

$$\alpha g_4(x_1, x_2, x_3) = x_2 + h(x_1, x_3) = 0 \tag{10}$$

where α is some operator.

Thus

$$x_2 = -h(x_1, x_3) \tag{11}$$

and the problem becomes one of choosing x_1 and x_3 to maximize

$$P = f(x_1, -h(x_1, x_3), x_3) \tag{12}$$

subject to the constraints

$$g_1(x_1, -h(x_1, x_3), x_3) \leq 0 \tag{13}$$

$$g_2(x_1, -h(x_1, x_3), x_3) \geq 0 \tag{14}$$

$$g_3(x_1, -h(x_1, x_3), x_3) \geq 0 \tag{15}$$

If the set of equalities cannot be solved for the variables seriatim, then a numerical procedure could be used. We do not recommend the replacement of equalities by inequalities as was done by Bracken and McCormick (1968). This will introduce errors into the solution, as will be shown in our example. The examples also show that usually the equality constraints present no problem.

Thus without any loss of generality, we consider inequality constraints only and formulate the general problem as follows:

Maximize

$$P = f(x_1, x_2, \ldots, x_n) \tag{16}$$

subject to the constraints

$$g_i(x_1, x_2, \ldots, x_n) \leq 0 \quad i = 1, 2, \ldots, m \tag{17}$$

$$h_j(x_1, x_2, \ldots, x_n) \geq 0 \quad j = 1, 2, \ldots, r \tag{18}$$

Suppose we take 100 random values for the set $\{x_1, \ldots, x_n\}$ and test each set with respect to inequalities (17) and (18). Two possible outcomes could result: none of the sets satify all the inequality constraints, or some sets (let us say 37) satisfy all the inequality constraints. If sufficient care is taken to restrict the range over which the variables $x_1, \ldots, x_n$ are chosen it may be quite reasonable to assume that the second outcome results. Then substituting these 37 sets of values into Equation (5) will give 37 values of P out of which the largest may be picked out. Let us denote the largest of these P by $P^{*(1)}$ and label the corresponding values of the variables $(x_1^{*(1)}, x_2^{*(1)}, \ldots, x_n^{*(1)})$. Next let us choose another 100 sets of $\{x_1, \ldots, x_n\}$, but restrict that the variables be chosen at random around the point $(x_1^{*(1)}, x_2^{*(1)}, \ldots, x_n^{*(1)})$ so that the probability that x_i is greater than $x_i^{*(1)}$ is the same as the probability that x_i is less than $x_i^{*(1)}$. Now if we test each of these sets with respect to Equation (17) and Equation (18), we may find many which satisfy the constraints, let us say 51. We may calculate 51 values for P, pick the maximum and denote it by $P^{*(2)}$. $P^{*(2)}$ may be greater than $P^{*(1)}$, but not necessarily. If we were to continue in this fashion no assurance could be given that the optimum will be found in a reasonable number of steps.

We may, however, increase the probability that $P^{*(2)}$ be greater than $P^{*(1)}$ if we reduce the range over which the values of x_i may be chosen after the first step. Suppose that initially the range is $x_i \pm 0.5$ and we decrease the range by 5% after each step. Then after 200 steps the range is $(.95)^{200} = 3.5 \times 10^{-5}$; that is, $x_i \pm 1.8 \times 10^{-5}$. This large effect should not be surprising to one who thinks of investing \$1 at 5% cumulative interest for 200 years. For each of the variables the initial range can be chosen according to the understanding one has for the problem, but one has the assurance that at 5% reduction the range is reduced to 3.5×10^{-5} of its initial value. Since the values for the variables are always chosen around the best point determined in the previous iteration, there is a good likelihood of convergence to the optimum. We therefore propose the following algorithm.

DIRECT SEARCH USING RANDOM NUMBERS COMBINED WITH INTERVAL REDUCTION ALGORITHM

1. Take initial values for $x_1, x_2, \ldots, x_n$ and an initial range for each variable; denote these by $x_1^{*(0)}, x_2^{*(0)}, \ldots, x_n^{*(0)}$ and by $r_1^{(0)}, r_2^{(0)}, \ldots, r_n^{(0)}$. Set the iteration index j to 1.

2. Read in a sufficient number of random numbers (let us say 2000) between -0.5 and $+0.5$. Denote these by y_{ki}.

3. Take pn random numbers from 2 and assign these to $x_1, x_2, \ldots, x_n$ so that we may have p sets of values, each calculated by

$$x_i^{(j)} = x_i^{*(j-1)} + y_{ki}\, r_i^{(j-1)} \quad i = 1, \ldots, n \qquad k = 1, \ldots, p$$

4. Test Equation (17) and Equation (18) and calculate a value of P with each admissible set.

5. Find the set which maximizes P given by Equation (16). Write out the maximum value of P and the corresponding $x_i^{*(j)}$, $i = 1, \ldots, n$. Increment j by 1 to $j + 1$.

6. If the number of iterations has reached the maximum allowed, end the problem. For example, we may choose 200 to be the maximum number of iterations.

7. Reduce the range by an amount ϵ

$$r_i^{(j)} = (1 - \epsilon) r_i^{(j-1)}, \quad \epsilon > 0$$

For example we may choose $\epsilon = 0.05$.

8. Go to step 2 and continue.

To illustrate the procedure and to test its effectiveness we choose 6 examples. In every example at each iteration we choose 100 sets of pseudo-random numbers in an interval -0.5 to 0.5, and the reduction rate of interval, ϵ, equal to 0.05, and continue the search until 200 iterations have been reached. We followed the standard modulus method to generate 2000 pseudo-random numbers in the range 0 to 999 and had them punched out by the computer in 20I4 format on 100 cards. Then by reading these data cards in format 20F4.3 we have numbers in range 0.000 to 0.999. Then by subtracting -0.5 from each number we have the numbers in range -0.5 to 0.5. This part of calculation is included in each problem so that the computation times which are reported may be

somewhat high. The computation time is reported for IBM 370/165 digital computer and is the total time required to execute the problem including the writing out of results.

Example 1: Optimum Fuel Allocation in Power Plants

Consider the problem of minimizing the purchase of fuel oil when it is desired to produce an output of 50 MW from a two-boiler-turbine-generator combination which can use fuel oil or blast furnace gas (BFG) or any combination of these. The maximum BFG that is available is specified. This problem corresponds to case 3 of Hovenessian and Stout (1963) and to case 2 of Hovanessian and Pipes (1969).

From Figure 6 of Hovanessian and Stout (1963) by applying nonlinear curve-fitting we obtained the fuel requirements for the two generators explicitly in terms of MW produced. For generator 1 we have the fuel requirements for fuel oil in tons per hour

$$f_1 = 1.4609 + .15186x_1 + .00145x_1^2 \qquad (19)$$

and for BFG in fuel units per hour

$$f_2 = 1.5742 + .1631x_1 + .001358x_1^2 \qquad (20)$$

where x_1 is the output in MW of generator 1. The range of operation of the generator is

$$18 \leq x_1 \leq 30 \qquad (21)$$

Similarly for generator 2 the requirement for fuel oil is

$$g_1 = .8008 + .2031x_2 + .000916x_2^2 \qquad (22)$$

and for BFG,

$$g_2 = .7266 + .2256x_2 + .000778x_2^2 \qquad (23)$$

where x_2 is the output in MW of generator 2. The range of operation of the second generator is

$$14 \leq x_2 \leq 25 \qquad (24)$$

It is assumed that only 10.0 fuel units of BFG are available each hour and that each generator may use any combination of fuel oil or BFG. It is further assumed that when a combination of fuel oil and BFG is used the effects are additive. That is, if in generator 1 we use fuel oil and BFG in 1/3 ratio to produce x_1 MW, then the total fuel consumption consists of 0.25 f_1 tons of fuel oil per hour and 0.75 f_2 fuel units of BFG per hour.

The problem is to produce 50 MW from the two generators in such a way that the amount of fuel oil consumed is minimum. Mathematically the formulation of the problem is as follows:

Minimize

$$C = x_3 f_1 + x_4 g_1 \qquad (25)$$

where f_1 and g_2 are given by Equations (19) and (22) subject to the following constraints:

(a) Operating range for the generator 1

$$18 \leq x_1 \leq 30 \qquad (26)$$

(b) Requirement of 50 MW of power

$$x_2 = 50 - x_1 \qquad (27)$$

(c) Operating range of generator 2

$$14 \leq x_2 \leq 25 \qquad (28)$$

(d) Fraction of fuel oil used in generator 1

$$0 \leq x_3 \leq 1 \qquad (29)$$

(e) Fraction of fuel oil used in generator 2

$$0 \leq x_4 \leq 1 \qquad (30)$$

(f) Availability of blast furnace gas (BFG)

$$\text{BFG} = (1 - x_3)f_2 + (1 - x_4)g_2 \leq 10.0 \qquad (31)$$

where f_2 and g_2 are given by Equations (20) and (23).

By using Equation (27), x_2 is eliminated and the problem is to choose the variables x_1, x_3, and x_4 so that C as given by Equation (25) is minimized. There are 9 inequality constraints embodied in Equation (26), and Equation (28) to (31). Note that there is no lower restriction on Equation (31) since computationally BFG cannot become negative.

The initial values for x_1, x_3, and x_4 were taken as 20, 0.5, and 0.5 and the initial regions, as 20, 1.0 and 1.0 respectively.

It took 1 second of computation time to perform 200 iterations after which the value of C was 3.05 with $x_1 = 30.00$, $x_2 = 20.00$, $x_3 = 0.00$, and $x_4 = 0.58$. In fact, these values were reached already at the 55th iteration.

Thus we see that the minimum fuel oil consumption is 3.05 tons/hr which is very close to the answer, 3.17 tons/hr, obtained by Hovanessian et al (1963, 1969) by means of separable programming where the nonlinearities were approximated by linear sections and the problem was solved by the standard linear programming procedure.

The most attractive feature of the direct search method is the simplicity of programming. Only the very basics of computer programming are required and it is not unlikely that the program will work perfectly the first time. Another advantage of the method is that the starting condition is not crucial—any reasonable value will do.

We now turn to an example where the answer is known exactly so that the procedure could be further checked.

Example 2: A Mathematical Problem with Known Solution

Consider the test problem introduced by Rosen and Suzuki (1965) and used by Gould (1971) to test his steepest descent procedure. The problem is a mathematical one which does not have any physical interpretation but has a unique optimum. The problem is to minimize

$$C = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4 \qquad (32)$$

subject to

$$x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_1 - x_2 + x_3 - x_4 - 8 \leq 0 \qquad (33)$$

$$x_1^2 + 2x_2^2 + x_3^2 + 2x_4^2 - x_1 - x_4 - 10 \leq 0 \qquad (34)$$

$$2x_1^2 + x_2^2 + x_3^2 + 2x_1 - x_2 - x_4 - 5 \leq 0 \qquad (35)$$

The minimum value of C is known to be

$$C = -44 \text{ at } x_1 = 0, x_2 = 1, x_3 = 2, x_4 = -1$$

Gould (1971) by using a gradient approach was able to obtain $x_1 = .0018$, $x_2 = 1.0073$, $x_3 = 2.0014$, $x_4 = -0.9967$ in 5 seconds of computation time on an IBM 360/75 computer.

By starting with $x_i = 0.0$ and $r_i = 0.5$ for $i = 1, 2, 3, 4$, we were able to obtain in 200 iterations requiring 2 seconds of computer time the following results: $C = -44.0000$, $x_1 = 0.0009$, $x_2 = 1.0004$, $x_3 = 1.9992$ and $x_4 = -1.0008$. We are thus able to reach much closer to the optimum than Gould.

To examine the effect of the choice of initial range, we also ran the program with $x_i = 0.0$ but $r_i = 5.0$ for $i = 1, 2, 3, 4$. After 200 iterations requiring again 2 sec-

onds of computer time, we obtained $C = -43.9999$, $x_1 = 0.0008$, $x_2 = 1.0014$, $x_3 = 1.9990$, and $x_4 = -1.0009$. The results are not as good as before but still considerably better than those of Gould (1971).

Example 3: Chemical Equilibrium in Complex Mixtures

We consider the chemical equilibrium problem formulated and solved by White et al. (1958) and by Dantzig et al. (1958) by means of piecewise linear approximations followed by linear programming. They also solved the problem by means of steepest descent. Here we solve the nonlinear programming problem by the proposed method.

It is known that at equilibrium the free energy is at a minimum. Therefore to determine the relative amounts of the species present in an equilibrium situation one can set up the problem of minimization of the free energy. By considering the N, H, O system of White et al. (1958) where there are 10 species present the problem is to minimize

$$C = \sum_{j=1}^{10} x_j \left[c_j + \ln\left(\frac{x_j}{x_T} \right) \right] \tag{36}$$

where x_T is the total moles and c_j are given by White et al. (1958). The constraints are

$$x_1 + 2x_2 + 2x_3 + x_6 + x_{10} = 2 \tag{37}$$

$$x_4 + 2x_5 + x_6 + x_7 = 1 \tag{38}$$

$$x_3 + x_7 + x_8 + 2x_9 + x_{10} = 1 \tag{39}$$

$$x_j \geq 0 \quad j = 1, 2, \ldots, 10$$

By using Equations (37) to (39) we may write

$$x_4 = 1 - 2x_5 - x_6 - x_7 \tag{40}$$

$$x_3 = 1 - x_7 - x_8 - 2x_9 - x_{10} \tag{41}$$

$$x_1 = 2 - 2x_2 - 2x_3 - x_6 - x_{10} \tag{42}$$

so that x_4, x_3, and x_1 can be eliminated and search be performed over the 7 remaining variables.

By starting with $x_i = 0.2$ and $r_i = 0.2$ for $i = 2, 5, 6, \ldots, 10$ (x_1, x_3, and x_4 are obtained from the equality constraints) we obtained after 200 iterations, which required 8 seconds of computer time, the minimum value for the free energy, $C = -47.7611$ and $x_1 = 0.04067$, $x_2 = 0.14774$, $x_3 = 0.78315$, $x_4 = 0.00141$, $x_5 = 0.48525$, $x_6 = 0.00069$, $x_7 = 0.02740$, $x_8 = 0.01795$, $x_9 = 0.03732$, and $x_{10} = 0.09686$. White et al. (1958), by using linear programming and steepest descent also obtained $C = -47.7611$. All the x_i we obtained agree identically to five decimal places with those of White et al. with the exception of three for which they obtained $x_2 = 0.14773$, $x_9 = 0.03731$ and $x_{10} = 0.09687$. Even here, the deviations are only in the last figure.

Example 4: Alkylation Process Optimization

Alkylation process is important in upgrading gasoline. We thus consider the problem of determining the best operating conditions in the alkylation process described by Payne (1958) and used for optimization by Sauer et al. (1964) who reduced the nonlinear problem to a series of linear programming problems. Since Bracken and McCormick (1968) formulated this problem as a direct nonlinear programming model but failed to obtain a correct solution to the problem, we shall use the same notation as Bracken and McCormick, but solve the problem by the proposed procedure.

Define

x_1 = olefin feed (barrels/day)
x_2 = isobutane recycle (barrels/day)
x_3 = acid addition rate (thousands of pounds/day)
x_4 = alkylate yield (barrels/day)
x_5 = isobutane makeup (barrels/day)
x_6 = acid strength (weight percent)
x_7 = motor octane number
x_8 = external isobutane-to-olefin ratio
x_9 = acid dilution factor
x_{10} = F-4 performance number.

When we use the data from Sauer et al. (1964), the problem may be formulated as follows:

Maximize

$$P = .063x_4x_7 - 5.04x_1 - .035x_2 - 10x_3 - 3.36x_5 \tag{36}$$

subject to the inequality constraints:

$$0.010 \leq x_1 \leq 2000 \tag{37}$$

$$0.010 \leq x_2 \leq 16000 \tag{38}$$

$$0.010 \leq x_3 \leq 120 \tag{39}$$

$$0.010 \leq x_4 \leq 5000 \tag{40}$$

$$0.010 \leq x_5 \leq 2000 \tag{41}$$

$$85 \leq x_6 \leq 93 \tag{42}$$

$$90 \leq x_7 \leq 95 \tag{43}$$

$$3 \leq x_8 \leq 12 \tag{44}$$

$$1.2 \leq x_9 \leq 4 \tag{45}$$

$$145 \leq x_{10} \leq 162 \tag{46}$$

and the equality constraints

$$x_4 = x_1(1.12 + .13167x_8 - .006667x_8^2) \tag{47}$$

$$x_5 = 1.22x_4 - x_1 \tag{48}$$

$$x_2 = x_1x_8 - x_5 \tag{49}$$

$$x_6 = 89 + [x_7 - (86.35 + 1.098x_8 - .038x_8^2)]/.325 \tag{50}$$

$$x_{10} = -133 + 3x_7 \tag{51}$$

$$x_9 = 35.82 - .222x_{10} \tag{52}$$

$$x_3 = .001x_4x_6x_9/(98 - x_6) \tag{53}$$

In this problem there are 7 equality constraints. Therefore, there are only 3 independent variables. We chose these to be x_1, x_7, and x_8. From these x_4, x_5, x_2, x_6, x_{10}, x_9, and x_3 can be calculated. The problem is therefore quite easy.

By taking $p = 100$ sets of random numbers per iteration and the initial range for x_1, x_7, and x_8 to be 500, 6 and 8 respectively, in double precision we required only 2 seconds of computer time to carry out 200 iterations. We arrive at the maximum P of 1162.0 with $x_1 = 1728.4$, $x_2 = 16000$, $x_3 = 98.4$, $x_4 = 3056.0$, $x_5 = 2000.0$, $x_6 = 90.6$, $x_7 = 94.2$, $x_8 = 10.41$, $x_9 = 2.61$, $x_{10} = 149.60$.

It will be noted that these values are very close to the results of Sauer et al. (1964). The slight deviation is present because the equalities used by Sauer et al. are only approximately satisfied. The reader might want to verify this point, but in any case our value of 1162.009 is very close to the value 1162.94 obtained by Sauer et al.

Example 5: Optimization of Drying Process for a Through-Circulation Dryer

The problem is to find the air flow rate and the bed thickness which will maximize the production rate. We

use the data and the corrected equations in the work of Chung (1972).

It is necessary to find the mass flow rate x_1 and the bed thickness x_2 such that the drying production rate is maximized; that is, maximize

$$P = 0.033x_1\left[\frac{0.036}{1-e^{-107.9x_2/x_1^{0.41}}}+0.095-\frac{9.27\times10^{-4}x_1^{0.41}}{x_2}\ln\left(\frac{1-e^{-5.39x_2/x_1^{0.41}}}{1-e^{-107.9x_2/x_1^{0.41}}}\right)\right]^{-1} \quad (54)$$

subject to the constraints

$$0.2-4.62\times10^{-10}x_1^{2.85}x_2-1.055\times10^{-4}x_1\geqq 0 \quad (55)$$

$$4/12-8.20\times10^{-7}x_1^{1.85}x_2-2.25/12\geqq 0 \quad (56)$$

$$2-109.6\frac{x_2}{x_1^{0.41}}\left[\frac{0.036}{1-e^{-107.9x_2/x_1^{0.41}}}+0.095-\frac{9.27\times10^{-4}x_1^{0.41}}{x_2}\ln\left(\frac{1-e^{-5.39x_2/x_1^{0.41}}}{1-e^{-107.9x_2/x_1^{0.41}}}\right)\right]\geqq 0 \quad (57)$$

With a starting point of $x_1 = 800$, $x_2 = 0.5$ and $r_1 = 200$, $r_2 = 0.25$ we arrive at the maximum P of 172.49 and $x_1 = 976.76$, $x_2 = 0.5235$. The computation time was 5 seconds. These values agree quite well with the corrected results of Chung (1973).

Example 6: Complex Reactor System Optimization

A very realistic chemical engineering problem has been introduced by Di Bella and Stevens (1965) and Adelman and Stevens (1972) who considered the optimization of a system consisting of a continuous stirred tank reactor, a heat exchanger, a decanter, and a distillation column. The design problem was earlier proposed by Williams and Otto (1960). To describe the system required 9 equations in 13 variables. The objective function to be maximized was per cent return on investment. Also an upper and lower constraint was placed on the reactor temperature.

Di Bella and Stevens (1965) formulated the optimization problem as a combination of steepest descent and linear programming but were unable to reach close to the optimum even after 600 passes through the routine. Adelman and Stevens (1972) used a constrained simplex method to arrive very close to the optimum. Luus and Jaakola (1973) showed that all the equations used for system description are not independent and there are 5 independent variables instead of 4. Therefore the system is greatly simplified.

The problem is to maximize the percent return given by

$$P = (84F_A - 201.96F_D - 336F_G + 1955.52F_P - 2.22F_R - 60V\rho)/(6V\rho) \quad (58)$$

subject to the constraints

$$F_G = 1.5k_3F_{RC}F_{RP}\beta \quad (59)$$

$$F_{RA} = \frac{F_{RC}}{10k_1F_{RB}}(10k_2F_{RB}+5k_3F_{RP}+k_2F_{RB}F_{RC}/b) \quad (60)$$

$$F_P = F_{RP} - b \quad (61)$$

$$F_D = 0.2\alpha\beta k_2F_{RB}F_{RC}/b \quad (62)$$

$$F_A = k_1F_{RA}F_{RB}\beta + F_DF_{RA}/\alpha \quad (63)$$

Table 1. Computational Requirements to Reach within 0.1% of the Optimum

Example no.	Performance index	No. of iterations	No. of function evaluations
1	3.053	55	1952
2	−43.962	28	1759
3	−47.714	17	394
4	1161.50	24	919
5	172.46	7	336
6	121.45	20	1244

Table 2. Computational Requirements to Reach within 0.01% of the Optimum

Example no.	Performance index	No. of iterations	No. of function evaluations
1	3.0526	86	2989
2	−43.996	61	3311
3	−47.757	48	1011
4	1161.90	53	1863
5	172.47	32	1323
6	121.53	41	2853

$$F_B = F_G + F_P + F_D - F_A \quad (64)$$

$$F_R = \alpha + F_G + F_P \quad (65)$$

$$F_{RE} = 10(F_{RP} - F_P) \quad (66)$$

where

$$\beta = V\rho/F_R^2 \quad (67)$$

$$b = F_{RP} - 0.2(4\beta k_2F_{RB}F_{RC} - 2.5k_3\beta F_{RC}F_{RP}) \quad (68)$$

$$\alpha = F_{RA} + F_{RB} + F_{RC} + 11b \quad (69)$$

For the purposes of optimization we consider as independent variables T, F_{RC}, F_{RP}, β, and F_{RB}.

By starting with $T = 650$, $F_{RC} = 7500$, $F_{RP} = 16000$, $\beta = 7.693\times10^{-9}$, $F_{RB} = 100000$, and the initial regions to be respectively 100, 3000, 9000, 6.154×10^{-9}, 50000, we obtained an optimum return of 121.534 within 200 iterations and 4 seconds of computer time. The variables are not uniquely determined at this value of the performance index as is shown by Luus and Jaakola (1973).

EVALUATION OF THE PROPOSED METHOD IN TERMS OF NUMBER OF FUNCTION EVALUATIONS

From the examples that have been provided it is clear that the optimum was always obtained with no more than 20000 function evaluations. Since the number of function evaluations cannot be determined *a priori*, we inserted a counting procedure into the programs to count the number of times the performance criterion function was evaluated. The cumulative number of function evaluations was printed out after every iteration.

In Tables 1 and 2 it can be seen that to reach within 0.01% of the optimum requires roughly three times as many function evaluations as to get within 0.1% of the optimum. Also, in every case we reached within 0.01% of the optimum in fewer than 100 iterations and fewer than 3000 function evaluations. It must be emphasized that no auxiliary functions have been introduced so it is difficult to compare the number of function evaluations to the number required for methods where auxiliary func-

tions such as gradients are required.

In spite of the difficulty of making a direct comparison of the proposed method to widely different optimization procedures, we solved some of the problems presented by Keefer (1973) for which the only information related to computational effort is the number of function evaluations.

To solve the very trivial first problem of Keefer required only 261 function evaluations by the proposed method compared to 691 required by Keefer. For the second problem involving a simplified version of alkylation problem, Keefer obtained convergence to within 2% of the optimum in 3487 function evaluations. As reported in Table 1, we obtained convergence to within 0.1% of the optimum in 919 function evaluations. The comparison for this problem is not direct since the equations used by Keefer have been simplified and Keefer unfortunately does not give the equations which he used.

For the third problem of Keefer (1973) the xylene separation and performance index equations are given by Gottfried et al. (1970). We solved the problem with the proposed method and reached to within 0.4% of the optimum in 73 iterations and 1170 function evaluations. (For starting conditions we chose $x_1 = 1500$, $x_j = 0.5$, $j = 2, \ldots, 7$, and $r_i^{(0)} = 0.5x_i^{(0)}$.) The computation time for 200 iterations was 6 seconds, and for 73 iterations we needed 2 seconds. The number of function evaluations 3345 reported by Keefer (1973) is considerably higher and the computation time of 3 min on IBM 360/75 computer as reported by Gottfried et al. (1970) is about 18 times higher if we allow a factor of 5 for the difference in computation time of IBM 370/165 versus 360/75. This particular problem shows that equality constraints where the variables cannot be solved seriatim in terms of other variables present no difficulties.

Unfortunately inadequate information is given by Keefer (1973) on the remaining three problems. Especially problems 5 and 6 would have been interesting since Keefer reported a larger number of equality constraints than the number of variables! Also for these problems the constraints are violated to cast further doubt on the validity of Keefer's results.

Recently, overestimation of the difficulty of solving nonlinear algebraic equations has given rise to numerous optimization procedures which have rather limited use for realistic problems. Such anticipated difficulties do not usually arise as is shown in some detail by Luus (1973). We do not want to delve into this problem here except to mention that the recently proposed method of Mamen and Mayne (1972) falls into this category. Their pseudo Newton-Raphson method for function minimization requires considerable effort already with the two very simple examples that were chosen. These examples can be solved readily by our proposed method or preferably by the standard method of Lagrange multipliers followed by the procedure described by Luus (1973). Due to the difficulty of obtaining convergence with the method of Mamen and Mayne (1972) even for simple problems, we do not attempt any comparisons to that method.

For problems where more than one local optimum can occur, the proposed method is especially useful. As an illustration we may look at the problem of maximizing

$$P = x_1^2 + x_2^2 + x_3^2$$

subject to the constraints

$$x_1 + 2x_2 + 3x_3 = 1$$

$$x_1^2 + \frac{x_2^2}{2} + \frac{x_3^2}{4} = 4$$

With the proposed method the global maximum $P = 9.995$ is obtained immediately, whereas the use of standard methods may give the local maximum $P = 9.051$ or either of the two minima $P = 4.430$ or $P = 4.049$.

In terms of the number of function evaluations the proposed method is very efficient. But its greatest usefulness is in the ease of programming and the reliability of the results obtained. With this method the constraints are always satisfied and the presence of other local optima usually presents no problem.

SUMMARY

We have presented a new method to solve nonlinear programming problems. Our experience in using this direct search method with a wide variety of problems indicates that it is a very fast and straightforward procedure.

For all the examples in this paper we have arbitrarily used 100 sets of values over which maximization or minimization is done in each iteration. We continued the process until 200 iterations had been performed. Also, we have arbitrarily set the rate of region reduction after each iteration at 0.05. Although for each problem there are optimum values for these parameters, the solution procedure is so fast that the choice is not really important in solving the problems of the type of complexity presented in this paper. If the problem is much more complicated and if the optimization procedure is to be used more than once, then it is natural to investigate the most effective choice of these parameters. This best choice, however, will be problem dependent.

ACKNOWLEDGMENT

This work was performed with the assistance of a grant from the Canadian National Research Council, A-3515. Computations were carried out with the facilities of the University of Toronto Computer Centre.

NOTATION

BFG = blast furnace gas
C = function to be minimized
f = function
g = function
h = function
i = index
j = index; as superscript j is used to denote iteration number
k = index; number of equality constraints
m = number of $\{\leq\}$ constraints
n = number of variables
p = number of sets of values taken at each iteration
P = function to be maximized
q = arbitrary function
r = number of $\{\geq\}$ constraints
r_i = region of ith variables
x_i = variables
α = operator
ϵ = reduction factor used for region reduction (for all examples here $\epsilon = 0.05$)

$$r_i^{(j+1)} = (1 - \epsilon) r_i^{(j)}$$

LITERATURE CITED

Adelman, A., and W. F. Stevens, "Process Optimization by the 'Complex' Method," *AIChE J.*, **18**, 20 (1972).

Bracken, J., and G. P. McCormick, *Selected Applications of Nonlinear Programming*, Wiley, New York (1968).

Chung, S. F., "Mathematical Model and Optimization of

Drying Process for a Through-Circulation Dryer", *Can. J. Chem. Eng.*, **50**, 657 (1972).
———., Letter to Editor, *Can. J. Chem. Eng.*, **51**, 262 (1973).
Dantzig, G., S. Johnson, and W. White, "A Linear Programming Approach to the Chemical Equilibrium Problem," *Management Science*, **5**, 38 (1958).
Di Bella, C. W., and W. F. Stevens, "Process Optimization by Nonlinear Programming," *Ind. Eng. Chem. Process Design Develop.*, **4**, 16 (1965).
Gottfried, B. S., P. R. Bruggink, and E. R. Harwood, "Chemical Process Optimization Using Penalty Functions," *ibid.*, **9**, 581 (1970).
Gould, F. J., "Nonlinear Pricing: Applications to Concave Programming," *Operations Research*, **19**, 1026 (1971).
Hovanessian, S. A., and L. A. Pipes, *Digital Computer Methods in Engineering*, McGraw-Hill, N. Y. (1969).
Hovanessian, S. A., and T. M. Stout, "Optimum Fuel Allocation in Power Plants," *Trans. IEEE Power Apparatus Syst.*, **82**, 329 (1963).
Keefer, D. L., "Simpat: Self-Bounding Direct Search Method for Optimization," *Ind. Eng. Chem. Process Design Develop.*, **12**, 92 (1973).
Luus, R., "On the Usefulness of the Penalty Function for Mathematical Programming Problems", *J. Optimization Theory Applic.*, in press.
———., and T. H. I. Jaakola, "A Direct Approach to Optimization of a Complex System", *AIChE J.*, **19**, 645 (1973).
Mamen, R., and D. Q. Mayne, "A Pseudo Newton-Raphson Method for Function Minimization", *J. Optimization Theory Applic.*, **10**, 263 (1972).
Payne, R. E., "Alkylation—What You Should Know About This Process", *Petroleum Refiner*, **37**, 316 (1958).
Rosen, J. B., and S. Suzuki, "Construction of Nonlinear Programming Test Problems," Communications of the ACM, **8**, 113 (1965).
Sauer, R. N., A. R. Colville, and C. W. Burwick, "Computer Points in the Way to More Profits", *Hydrocarbon Processing Petrol. Refiner*, **43**, 84 (1964).
White, W. B., S. M. Johnson, and G. B. Dantzig, "Chemical Equilibrium in Complex Mixtures," *J. Chem. Phys.*, **28**, 751 (1958).
Williams, T. J., and R. E. Otto, "A Generalized Chemical Processing Model for the Investigation of Computer Control, *AIEE Trans.*, **79**, 458 (1960).

Manuscript received December 15, 1972; revision received March 21 and accepted April 5, 1973.

Membrane Ultrafiltration of a Nonionic Surfactant

Aqueous solutions 1.65×10^{-4} to 1.10×10^{-2} molar in the nonionic surfactant Triton X-100 are subjected to membrane ultrafiltration over the temperature range of 22 to 50°C in a continuous flow, thin channel cell, with channel Reynolds numbers from 50 to 1175. Data for the ultrafiltrate flux through the optimum polyelectrolyte complex membrane (of the three tested) are correlated by an equation involving the in-series resistances of the membrane and of the polarized boundary layer of the surfactant rejected by the membrane, with the latter resistance varied over a wide range. The correlations are used to estimate the required membrane area and resultant ultrafiltrate concentration of surfactant to achieve any specified water recovery using cells-in-series and cells-in-parallel operation. The gelation concentration and surfactant mass transfer coefficient are used to estimate water flux in the gel-polarization region.

R. B. GRIEVES
D. BHATTACHARYYA
W. G. SCHOMP
and
J. L. BEWLEY

University of Kentucky
Lexington, Kentucky 40506

SCOPE

Membrane ultrafiltration has been used successfully to concentrate or purify macromolecular solutes and colloidal particulates and for the treatment of aqueous wastes. In a recent series of articles, Porter and Michaels (1971a, b, c, d, 1972) have reviewed the applications of ultrafiltration in many processing areas. Ultrafiltration (in contrast to reverse osmosis) is carried out at low pressures of about 10^5 to 10^6 N/m^2 and involves the separation of modest molecular weight (500 and greater) solutes, macromolecules, and colloids from aqueous solution, utilizing comparatively-open, diffusive (or microporous) anisotropic membranes. However, severe performance limitations are encountered during ultrafiltration due to the high-flux characteristics of the membranes, which result in the