Date of Issue: May 30, 1968

Report Number: K-1746

Subject Category: Mathematics and Computers

A GRADIENT PROJECTION METHOD FOR CONSTRAINED OPTIMIZATION

Kenneth E. Cross

Applied Science Department

Computing Technology Center

UNION CARBIDE CORPORATION
NUCLEAR DIVISION
Computing Technology Center
Oak Ridge, Tennessee

ACKNOWLEDGMENTS

The author wishes to express his gratitude to Messrs. J. R. Copper, T. E. Pritchard, and W. L. Kephart for their helpful discussions about the field of nonlinear optimization. The editorial advice given by Messrs. L. L. Anthony and N. H. Van Wie is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF FIGURES

A GRADIENT PROJECTION METHOD FOR CONSTRAINED OPTIMIZATION

ABSTRACT

A description is made of the constraint portion of a nonlinear optimization program which was developed for the purpose of solving a series of engineering design and economic problems. Additional capabilities were incorporated into an existing gradient search program to enable the consideration of both linear and nonlinear, equality and inequality, constraints. A rather nontheoretical approach is attempted in describing both the philosophy and the mechanics involved in the constraint techniques.

I. INTRODUCTION

In the field of nonlinear optimization there are many methods [1,2] which may be used successfully on unconstrained problems. However, most practical problems arising in industry involve constraints, both general and specific, which must be satisfied in order to obtain meaningful results. The technique described here was developed for the purpose of optimizing a series of constrained nonlinear problems utilizing an existing nonlinear optimization program.[3] This particular program uses a gradient search procedure, with refinements for recognizing patterns in the response surfaces and boundaries on the variables. However, the constraint technique should be applicable to most gradient search programs since they operate within the feasible solution space while searching for an optimum. The goal of the constraint technique is the projection of gradient so as to obtain a feasible direction and the adjustment of points which fail to satisfy the constraints. Designed with generality and flexibility in mind, this code has been used to solve a wide variety of engineering and management problems in which the function to be optimized was relatively smooth, yet involved both linear and nonlinear constraints.

II. STATEMENT OF THE PROBLEM

The basic problem is the optimization (maximization or minimization) of a function of n variables:

$$F = f(x_1, x_2, x_3, \ldots, x_n),$$

subject to m constraints:

$$g_i(x_1, x_2, x_3 \ldots x_n) \geqq 0, \text{ where } i = 1, 2, 3, \ldots m.$$

Since most techniques for projecting onto constraints are designed for equality constraints, the problem of handling inequality constraints arises. Use of slack variables, as found in linear programming, has been suggested. However, this alternative has several disadvantages. First, the number of variables under consideration in the gradient is increased. In addition, the inequality constraint is merely replaced by a lower bound on the slack variable. A last, but major, objection to the use of slack variables is the necessity for considering all constraints at all times. In many problems there are inequality constraints which are not violated, and thus, need not be considered. The constraint technique to be described disregards inequality constraints until the search procedure attempts to violate them. Further, upper and lower bounds on the variables are considered to be inequality constraints when projecting the gradient. With these facts in mind, slack factors appear to present no significant improvement to the problem of constraint manipulation.

III. GRADIENT PROJECTION

The basic method of handling constraints is use of Lagrange multipliers to project a vector onto the constrained subspace of interest.[4] By definition, in an unconstrained region, n variables exist in an n-dimensional domain. However, when m equality constraints are applied, the dimension of the space is reduced by m. Lagrange multipliers are coefficients λ_i, $(i = 1, 2, \ldots m)$, applied to the partial derivatives of the equality constraints with respect to the n variables. This application results in a vector which adjusts the gradient to lie in an L-dimensional $(L = n\text{-}m)$ subspace. The L, or constrained, subspace is defined as the feasible area in which the m equality constraints are satisfied.

For example, let P be the n-dimensional column vector representing the gradient:

$$P = \nabla f$$

$$P_i = \partial f/\partial x_i;\ i = 1, 2, \ldots n. \tag{1}$$

Let G be the m by n matrix of partial derivatives of the constraints:

$$G_{ji} = \partial g_j/\partial x_i;\ i = 1, 2, \ldots n, \text{ and } j = 1, 2, \ldots m. \tag{2}$$

The Lagrange multipliers are the elements λ_i of an m-dimensional column vector:

$$\lambda = (GG^T)^{-1}(GP) \tag{3}$$

Using this vector, one can compute a new direction, PP, by adjusting P:

$$PP = P - G^T \lambda \tag{4}$$

This new direction, PP, will lie along the constraints themselves if they are linear. However, with nonlinear constraints the new gradient will lie along the hyperplanes, tangent to the constraints at the point under consideration.

This technique is a very precise projection method for satisfying equality constraints. However, the problem remains of identifying which inequality constraints to include in the projection (that is, which inequalities to consider as equalities).

IV. VIOLATION OF CONSTRAINTS

After a feasible direction is determined and steps, which improve the objective function, have been taken in that direction, further progress may result in violation of an inequality constraint. Therefore, a method of accepting only feasible points as candidates for improving the objective function was devised.

When the violation involves a specified upper or lower bound on a variable, the most direct procedure is reduction of the step size. The length of the step is adjusted by a ratio: the distance between the last point and the bound, divided by the distance the variable actually moved. Should more than one variable violate a boundary, the step is scaled small enough to avoid the most imminent and, therefore, all violations. Although this procedure is precise for linear constraints, it loses accuracy when the constraints are nonlinear. Therefore, a linear projection technique as developed below is used when constraints other than boundaries are violated. One projection is sufficient for linear constraints, whereas with nonlinear constraints several projections may be required to adjust a point onto the constraints.

Let the m by n matrix, G, be as previously described:

$$G_{ji} = \partial g_j/\partial x_i;\ i = 1, 2, \ldots n, \text{ and } j = 1, 2, \ldots m.$$

Let ΔX be an n-dimensional column vector representing the necessary change in the variables in order to satisfy the constraints being considered.

Further, let Δg be an m-dimensional column vector representing the change in the constraints necessary for their satisfaction. For example, the constraint g_i should equal 0. If $g_i = C$, then $\Delta g_i = -C$. These changes are a function of the changes in the variables:

$$\Delta g = G(\Delta X) \tag{5}$$

An m-dimensional column vector, γ, whose elements are to be applied to the elements of G in order to obtain the desired ΔX, is then defined:

$$\Delta X = G^T(\gamma) \tag{6}$$

Substituting for ΔX in equation (5):

$$\Delta g = GG^T(\gamma).$$

Since Δg and G are known, one can solve for γ. Then using equation (6), ΔX can be determined.

For nonlinear constraints several iterations through this procedure may be necessary. However, most constraint violations are minor, and the procedure converges rapidly to a feasible point.

It should be noted that in the case of nonlinear constraints, the elements of the G matrix are approximations to the actual partial derivatives of the constraints with respect to the variables. This approximation is a result of calculating the partial derivatives using a point which does not lie on a constraint. In cases where constraints are almost linear, or where a constraint violation is small, the approximations in G are very good.

In rare cases a step may be taken by the search procedure which violates constraints to such an extent that the iterative procedure is unable to converge and give a feasible point. Such a situation might result from violating a constraint by an extremely large amount, or from an attempt to project onto too many constraints. These problems, however, are easily eliminated. After a predetermined number of iterations have failed to give a feasible point (within some tolerance of the constraints), the step is rejected. The program then determines a new gradient and reduces the step size in order to obtain a feasible point. Further progress will depend on the particular gradient search procedure with which the projection technique is used.

V. BOUNDARIES APPLICABLE TO VARIABLES

As previously stated, variable boundaries are treated differently than the inequality constraints already described. Bounds are linear inequality constraints, but involve only one variable. When a bound is exceeded, the length of the step is merely reduced to obtain a feasible point. However, when a bound is considered an inequality constraint onto which a projection will be made, its treatment is different. For example, if the variable x_i is to be held at one of its bounds, $x_i = a$, $\partial f/\partial x_i$ is set equal to zero. It is possible, however, that $\partial f/\partial x_i$ will not remain zero when the gradient is projected onto other constraints. To assure a value of zero for this partial derivative, the variable x_i must be removed from consideration when projecting the gradient. This is accomplished by setting $\partial g_j/\partial x_i$ equal to zero for each constraint j being considered as an equality constraint during projection of the gradient. The projection can now be performed in a subspace in which the x_i direction has been eliminated from both the gradient and the constraints.

When any of the inequality constraints are nonlinear, the procedure just described may not produce the desired result. For instance, if a constraint becomes parallel and coincides with a boundary, all nonzero partial derivatives of a constraint g_j may be set equal to zero. Should this occur, the matrix GG^T will be singular, since all elements of row and column j will be zero. This singularity is merely a computational problem which can be eliminated by replacing the j^{th} diagonal element with one, whenever it is equal to zero. Since the j^{th} element of the vector GP is also zero, λ_j will be calculated as zero -- in effect disregarding the j^{th} constraint.

A more frequent cause of singularity in the matrix discussed above is the addition of another bound to the set already considered to be the set of inequality constraints. In this case the problem may be "over constrained," with the matrix GG^T being singular. Again, if the $j^{\underline{th}}$ row and column are zero vectors, setting the $j^{\underline{th}}$ diagonal element at 1 will force λ_j to be zero. If a step in the direction determined does not show improvement, or if the projection of the gradient is zero, a new gradient is calculated and projected onto the necessary inequality constraints.

VI. INEQUALITIES CONSIDERED AS EQUALITIES

The basic idea when projecting onto inequality constraints is consideration of as few constraints as possible. This is directly opposed to the linear programming philosophy whereby a maximum number of inequality constraints are treated as equalities.

After a new gradient is calculated, it is projected onto the intersection of the constraints under consideration. This set originally consists of equality constraints. To this are added the inequality constraints violated by the gradient calculation.

The dot product of the gradient and the unit normal is calculated for each inequality constraint to statisfy the above criteria. The constraint, corresponding to the normal resulting in the largest dot product, is added to the original set of equality constraints. This procedure is repeated until the gradient projection onto the constraint intersection results in a feasible direction.

In a simple two-variable, two-inequality constraint problem, a feasible direction will be found unless the gradient lies between the normals N_1 and N_2. In Figure 1, G_2 will be projected onto C_2 first, since $G_2 \cdot N_1 < G_2 \cdot N_2$. Then, since G_{2p} violates C_1, G_2 is projected onto the intersection of C_1 and C_2, thus obtaining a zero vector and, for this particular two-variable problem, an optimum. In the general case, when no successful steps can be taken in a particular direction, a new gradient is calculated, the inequality constraints are considered as equalities, and the search continues.

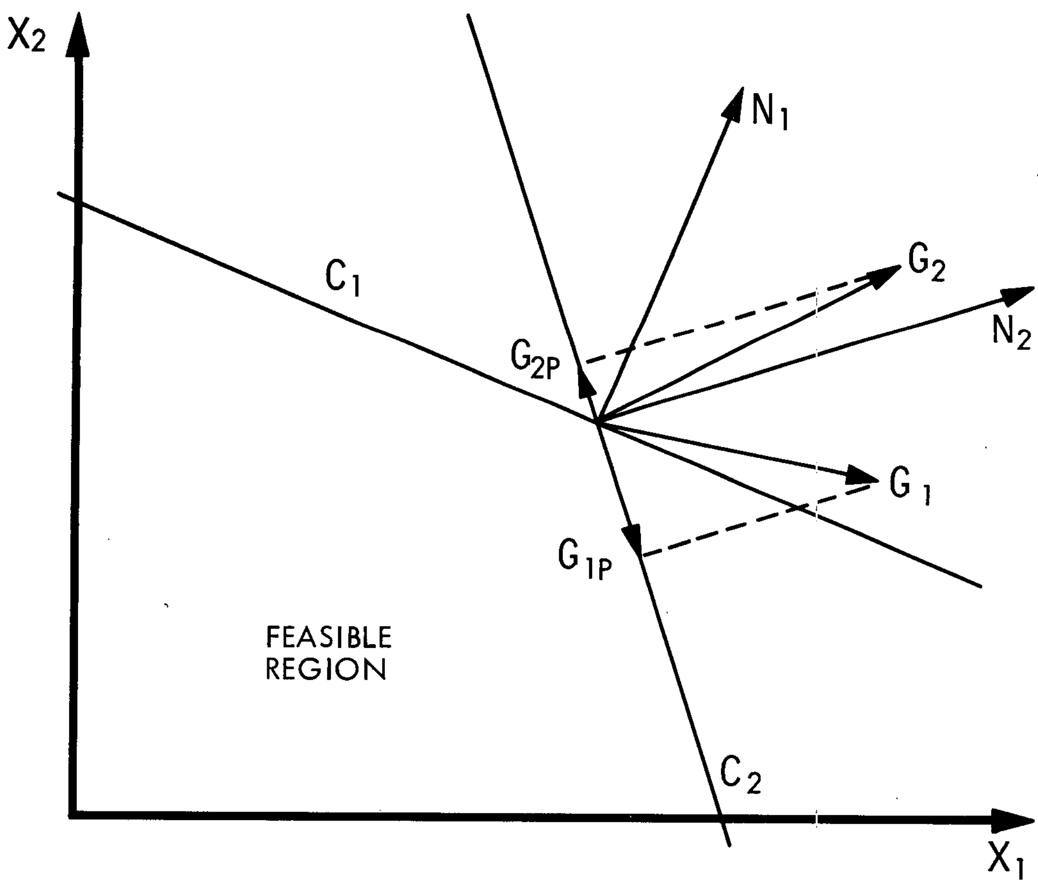

Figure 1. Gradient Projection at Optimum.

If, while moving in the direction determined, another inequality constraint is violated, this constraint is added to the set, and the gradient is again projected. Removal of constraints from the group considered as equalities occurs only when a new gradient is calculated. In each gradient calculation a minimum number of constraints is considered. Rarely will an excessive number of constraints be added during the search procedure. However, should the gradient projection fail due to an excess of constraint violations, a new gradient is calculated; and a minimum number of constraints is again considered. The search continues, provided an optimum has not been reached. The criterion for optimality is embedded within the particular gradient search procedure with which the project technique is associated. In this program two consecutive gradients with no successful steps signal an end to the search.

Following are two important points to consider when projecting a gradient. First, the gradient must be projected to satisfy all equality constraints. Further, when a gradient points in a direction which will violate an inequality constraint, and when the current point is near (within specified tolerance) the constraint, the gradient must be adjusted away from this infeasible direction. Although a gradient may violate several inequality constraints, it is not always necessary to project onto the intersection of all the violated constraints. In Figure 2 it is only necessary to project onto C_2 to obtain a feasible direction G_p.

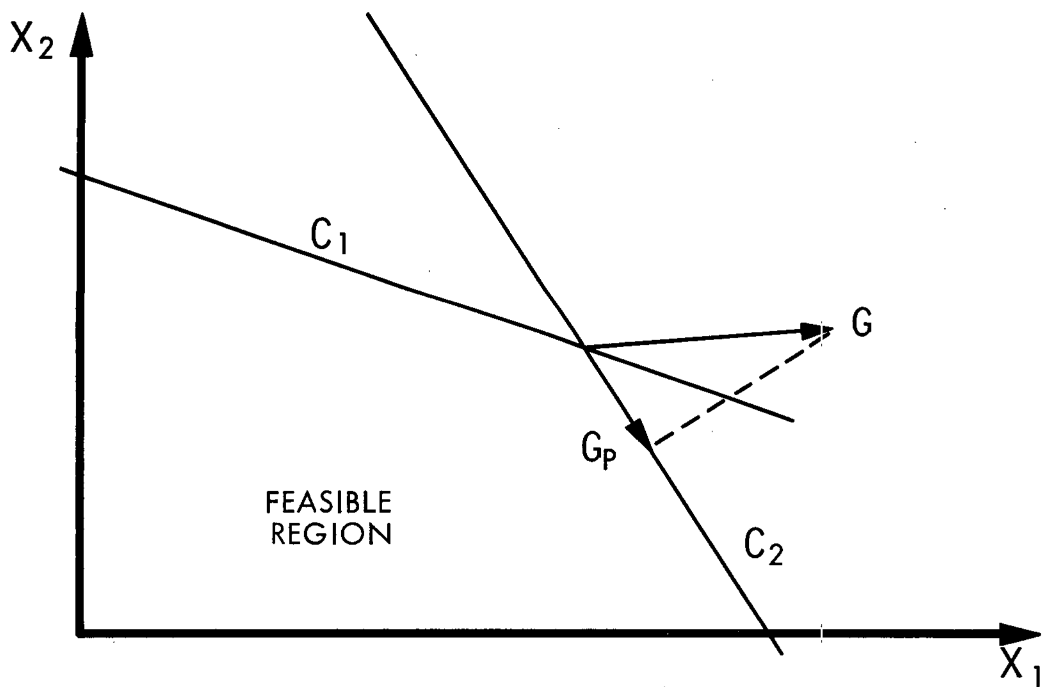

Figure 2. Gradient Projection at Nonoptimum Intersection of Constraints.

VII. CONCLUSION

The technique and the program into which it was incorporated are designed to handle general nonlinear problems with nonlinear constraints. The program is not intended to handle expeditiously any particular class of problems, such as those with steep ridges or other complicated surfaces. The method is intended to work with consistency on a variety of problems, even though in many cases slower solution times may result. The program may be characterized by the fact that it works best on small, relatively nonlinear problems.

The program has been used to solve problems with up to 50 variables and 50 constraints; however, it was intended primarily for use with fewer variables and constraints. Calculation of the objective function in problems that have been solved has varied from one arithmetic statement to a model represented by several hundred FORTRAN statements. The classes of problems solved vary from mathematical test problems to economic, design, and production models for gaseous diffusion, desalination, and nuclear reactor operations.

The program was compared with about 30 other nonlinear optimization codes, and the results were reported by A. R. Colville at the Mathematical Programming Symposium at Princeton University on August 14-18, 1967. The algorithm, which utilizes numerical approximation to derivatives, compared very favorably to codes which took advantage of the property of analytical first or second partial derivatives, which were present in most of the problems in the specified test set.

With this program, as with most nonlinear programs, the results obtained on real-life problems are very dependent on the design of the problem to be solved, as well as the effectiveness of minor adjustments of an algorithm to obtain best results for specific unusual problems.

The program is available as SHARE Release No. SDA 3541.

REFERENCES

1. Wilde, D. J., Optimum Seeking Methods, Prentice Hall, 1964.

2. Fletcher, R., and Powell, M.J.D., "A Rapidly Convergent Descent Method for Minimization," Comput. J. 6(1963), 163-168.

3. Copper, J. R., and Kephart, W. L., "Some Improvements in the Gradient Search Method of Optimization," Union Carbide Corporation, Nuclear Division, December 1965, Report K-DP-1063.

4. Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming," SIAM, V. 8 (1960), V. 9 (1961).