Mathematical Programming 3 (1972) 124–128. North-Holland Publishing Company

A NEW ALGORITHM FOR OPTIMIZATION

ANONYMOUS *

Received 18 May 1972

A procedure is presented for solving problems of a specified kind.

1. Introduction

Existing methods for solving the optimization problem of finding the maximum of a function f over a domain S have the following deficiencies:

(1). Special requirements are often placed on the nature of the function, such as differentiability or concavity.

(2). The domain is usually required to be connected, and often even convex.

(3). The procedures themselves are often complicated, making them difficult to understand and to program for a computer.

Accordingly, I and the associates referred to below undertook the development of a procedure which would be free of these limitations.

2. Mathematical background

Let S be a subset of a separable metric space U. As is well known, there exists a sequence $\{P_0, P_1, \dots\}$ of points of U which is dense in U. If S has the property (A): $S \subseteq$ closure (interior (S)), then $\{P_0, P_1, \dots\}$

* Editor's note: This manuscript was transmitted, torn and tattered, to Mathematical Programming by Philip Wolfe with a letter, stating, in part: "I have refereed many papers which proposed optimization algorithms without studying their effectiveness; it will save me much time to have here a single reference I can cite, saying 'This algorithm solves all the problems yours does and, on the available evidence, equally well.' I therefore recommend publication ... and hope that the author will come forward to receive ... what he richly deserves."

is also dense in S, and if f is continuous on S, then

$$\begin{aligned}\operatorname{Sup}\{f(x)\colon x \in S\} &= \operatorname{Sup}\{f(x)\colon x \in S\{P_0, P_1, \dots\}\} \\ &= \operatorname{Sup}_k\{f(P_k)\colon P_k \in S\}\ .\end{aligned}$$

We have thus been able to replace the maximization of f over the uncountable domain S by a maximization problem over a much smaller domain.

3. The algorithm

Let f, S, and the sequence $\{P_k\}$ be as in Section 2.
Initially set $j = 0$, $k = 0$, $x_0 = P_0$.
At step k, there are given $j \leq k$, x_j, and P_k.

(i) If P_k is not in S, replace k by $k+1$ and go to (i). Otherwise, go to (ii).

(ii) $[P_k \in S]$. If $f(P_k) > f(x_j)$, define $x_{j+1} = P_k$, replace k by $k+1$, and go to (i); otherwise replace k by $k+1$ and go to (i).

(In this form, the procedure is known in the literature as Wolfe's Universal Algorithm [1].)

4. Convergence of the algorithm

Theorem. Under the conditions of Section 2, the sequence $\{f(x_j)\}$ converges monotonically to the supremum of f on S.

Proof. The proof follows immediately from a classical result [2]. Recently a proof using more advanced techniques has been given [3].

5. Programming the algorithm

For convenience in programming the algorithm for automatic computation, it was decided to further specialize the space U to the Hilbert space of all square-summable sequences of real numbers. The resulting Separable Hilbert iterative technique allows the sequence $\{P_k\}$ to be generated internally instead of being read in from an external medium. The members are generated in blocks indexed by $N = 1, 2, \ldots,$ the Mth element of block N being the vector $(a_1, a_2, \ldots, a_N, 0, 0, \ldots)$, where $(a_1, \ldots, a_N)$ is the Mth point in the lexicographic ordering of the vertices of the subdivision into hypercubes of side $1/N$ of the N-cube of side N centered about the origin. (The fact that P_k thus has only finitely many nonzeroes is computationally convenient.) Of course, if the problem is of finite dimensions, say D, then all components of P_k after the Dth can be ignored. Our program in APL/360 is:

```
      ∇ SEPARABLEHILBERTITERATIVETECHNIQUE
[1]     N←0
[2]     X←(N←N+2+M←¯1)ρ0
[3]    S:→(S-M≥¯1+(1+N*2)*N),X←(X×~V)+P×V←(G P)∧(F X)<F P
        ←(¯0.5×N)+((Nρ1+N*2)⊤M←M+1):N
      ∇
```

The user need only establish two user-defined functions F and G: The function F must give the value of the objective function f, and the function G is to have the value 1 or 0 according as its argument belongs to the set S or not.

Our program establishes the workability of the new algorithm, but its effectiveness could be even further improved by taking advantage of its adaptability to parallel computation. Accordingly, the procedure has been reprogrammed to run on a computer having up to T parallel arithmetic registers. (The codes for this are unfortunately too numerous to be reproduced here.) It can be shown that, for suitable choice of T, the running time can be decreased by a substantial factor.

6. Refinements and extensions

Research is presently under way on making the algorithm even more effective for problems of optimization. It has been shown that when the set S is countable, it is sufficient to take $\{P_k\} \supseteq S$ to obtain the conclusion of the convergence theorem of Section 4, thus extending the algorithm to cases in which the hypotheses of that theorem do not hold, as when S consists of all the vectors in E^D whose components are integer valued (called "integer programming problems" [4]). It should be noted that the algorithm as programmed in Section 5 accomplishes exactly that. Such extensions will be the subject of forthcoming papers.

I am indebted to my colleague for pointing out the possibility of an intriguing acceleration device for the algorithm. In most problems it will be the case that the sequence $\{P_0, P_2, P_4, ...\}$ is dense in the sequence $\{P_0, P_1, P_2, ...\}$, from which it can be shown that the former will be dense in U. Thus if the sequence $\{P_k\}$ in the algorithm of Section 2 is replaced by the sequence $\{P_{2k}\}$, the convergence theorem still obtains, yet the resulting procedure requires 50% fewer steps than the original procedure. This finding has suggested certain other acceleration devices which seem to promise further substantial gains; they will be reported in due course in the literature.

7. Validating the algorithm

Since many procedures for the solution of optimization and other problems are known, we have made a careful comparison of the new method with existing algorithms. A T-register parallel processor not being available to us, we asked the associate mentioned above to undertake the simulation of the algorithm on such a computer by hand. He has reported very satisfactory progress in solving the problem [4] used to validate optimization algorithms.

As an independent validation, we have compared our algorithm with 43 published algorithms for optimization on all capabilities of major importance:

	Published algorithms	Our algorithm
Handles nondifferentiable function	13% "yes"	yes
Handles nonconcave function	33	yes
Handles integer variables	28	yes
Handles nonconvex domain	35	yes
Simple	8	yes
Programmed for computer	39	yes
Tested on the problem	38	yes
Reported very satisfactory	92	yes
Convergence proven	22	yes
Convergence rate established	–	–

(The absence of data in the last category is due to the fact that nothing [1] is known about convergence rates of algorithms, which will be the subject of a subsequent paper.)

It is clear that the new algorithm is superior to existing ones as measured by the standard criteria for publication; indeed, it has a score of 9 out of a possible 10, as against an average of 3.5, and a maximum of 8, for the test group.

8. Conclusions

Observations concerning the subject matter of the paper have been presented.

Acknowledgment

The author wishes to acknowledge the extent to which he has been supported in this work.

References

[1] Anon., On Wolfe's universal algorithm, Thesis, to appear.
[2] K.F. Gauss, *Gesammelte Werke* (Springer Verlag, Berlin 1933) (and other works by the same author).
[3] N. Bourbaki, *Eléments de mathématique* (Hermann et Cie., Paris) p. 1 et seq.
[4] Private Communication.