Use of a Monte Carlo Method in an Algorithm
Which Solves a Set of Functional Inequalities

A. J. HEUNIS

Communicated by P. P. Varaiya

Abstract. This paper shows how an existing algorithm, which is used in computer-aided design problems for solving a set of functional inequalities, may be modified by the inclusion of a Monte Carlo method to give a stochastic algorithm, which is easier to implement than its deterministic original and which solves the given set of functional inequalities almost surely.

Key Words. Functional inequalities, approximate maximization, uniform scattering of points, probability triple, almost sure convergence.

1. Introduction

In designing an engineering system to satisfy a given set of specifications, it is often necessary to solve the following mathematical problem:

Given jointly continuous functions $\phi_k: \mathbb{R}^n \times \Gamma_k \to \mathbb{R}$, $k = 1, \ldots, m$, where each $\Gamma_k$ is a compact subset of $\mathbb{R}^{p_k}$, and where, for each $k$ and $v_k$ in $\Gamma_k$, the function $x \mapsto \phi_k(x, v_k)$ is continuously differentiable on $\mathbb{R}^n$, compute a point $\hat{x}$ which belongs to the set of points

$$F \triangleq \{ x ; \phi_k(x, v_k) \leq 0, \text{for all } v_k \text{ in } \Gamma_k, k = 1, \ldots, m \}.$$

Several methods have been developed to solve this problem (Refs. 1 and 2), and one of the most efficient, given in Ref. 1, is as follows.

Algorithm 1.1. Specify a double indexed sequence $\{ \epsilon_{ij}, i \geq j \}$ of real positive numbers such that $\epsilon_{ij} \uparrow \bar{\epsilon}_j$, as $i \to \infty$, uniformly with respect to $j$, and

$$\lim_{j} \bar{\epsilon}_j = 0.$$

Step 0. Choose an $x_0$ in $\mathbb{R}^n$, and set $i = 0$. 

Step 1. For each \( k = 1, \ldots, m \), compute a \( v_{k,i} \) in \( \Gamma_k \) such that
\[
\phi_{k,i} \triangleq \phi_k(x, v_{k,i}) \text{ approximates } \max \{ \phi_k(x, v_k); v_k \in \Gamma_k \}.
\]
Store \( \phi_{k,i} \) and \( v_{k,i} \).

Step 2. For each \( k = 1, \ldots, m \), define
\[
\hat{\Gamma}_{k,i+1} = \{ v_{k,j}; \phi_{k,j} > \varepsilon_{ij} \text{ for } j = 1, \ldots, i \},
\]
and compute \( x_{i+1} \), which belongs to the set
\[
\{ x; \phi_k(x, v_k) \leq 0, \text{ for all } v_k \in \hat{\Gamma}_{k,i+1}, k = 1, \ldots, m \}.
\]

Step 3. Set \( i = i + 1 \), and go to Step 1.

Associated with this algorithm is the following convergence theorem, which has been established in Ref. 1.

**Theorem 1.1.** If the sequence \( \{ x_i \} \) generated by the algorithm contains a subsequence \( \{ x_{i}, i \in K \} \) which converges to \( \hat{x} \), and if, for each \( k = 1, \ldots, m \), it is true that
\[
\lim_{i \to \infty} \max_{i \in K} \{ \phi_k(x, v_k); v_k \in \Gamma_k \} = 0,
\]
then \( \hat{x} \in F \).

The practical implementation of this algorithm requires further methods for carrying out Steps 1 and 2. The computation of \( x_{i+1} \) in Step 2 involves finding a point in a set which is specified by a finite number of inequalities (since each of the sets \( \hat{\Gamma}_{k,i+1} \) is finite). This presents no problem, as algorithms have been developed (Refs. 3 and 4) for solving finite sets of inequalities and for doing so in a finite number of iterations.

The approximate maximizations in Step 1 are more troublesome, however, and are usually evaluated using the simplest possible techniques. These generally take the form of setting up \( r_i \) evenly spaced points in each \( \Gamma_k \), at the \( i \)th iteration, and then evaluating the maximum of \( \phi(x, v) \) over these points. If the integers \( r_i \) increase monotonically to infinity as \( i \to \infty \), then it follows from the joint continuity of \( \phi_k(\cdot, \cdot) \) that approximate maxima which satisfy the requirements of the above convergence theorem are obtained. However, it becomes difficult to construct evenly spaced points in sets \( \Gamma_k \) which are solid in linear spaces of high dimension. A way of avoiding this difficulty is to uniformly scatter \( r_i \) points in the sets \( \Gamma_k \) by suitably using a random number generator and then maximizing \( \phi_k(x, \cdot) \) over these randomly scattered points. Fortunately, uniform scattering of points is easy to obtain when the sets \( \Gamma_k \) are of regular form (e.g., ellipsoids),