

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:

The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

A NEW MINI-MAX, CONSTRAINED OPTIMIZATION
METHOD FOR SOLVING WORST CASE PROBLEMS

by

S.W. Director , L.M. Vidigal .
R.K. Brayton** & G.P. Hachtel**

DRC-18-2-79

January 1979

* Department of Electrical Engineering
Carnegie-Melion University
Pittsburgh, PA 15213

** T-J. Watson Research Lab
IBM
Yorktown Heights, NY

This research was sponsored in part by NSF Grant ENG-77-20895.

ABSTRACT

A new worst case design procedure is described. This method employs Powell's new constrained optimization procedure and is at least superlinearly convergent. A novel function splitting scheme is described to avoid singularity problems inherent in some previously reported methods.

I. Introduction

In this paper we describe a new mini-max based method for solving both the fixed and variable tolerance worst case problems. While this method is similar to the technique proposed recently by Madsen and Schjaer-Jacobsen [1], it is at least superlinearly if not quadratically convergent even for the non regular² case. Furthermore, the present method solves the variable tolerance problem directly rather than as a double iteration as given in [1].

¹ This research was sponsored in part by NSF Grant ENG 77 - 20895.

² We are using the term regular and singular as defined for a minr-max problem in [2], i.e., a minimax problem is singular if the rank of the Jacobian of the active constraint at the solution is less than n , the dimension of the space.

The fixed tolerance problem can be stated as follows. Given the tolerance t find the nominal point x^* such that the largest value of the constraints $f_i(x)$, $i=1, 2, \dots, m$ is minimized. Formally we can state this problem as

«

$$\begin{aligned} \min_x \quad & \max_{y \in \bar{R}_t} C_3(y) \\ & j = 1, 2, \dots, m \end{aligned} \quad (1)$$

where $R_t = \{y | x_i - t \leq y_i \leq x_i + t\}$ is the tolerance region. By introducing an auxiliary variable, this mini-max problem can be transformed into the constrained optimization problem.

$$\begin{aligned} \min \quad & y \\ \text{subject to} \quad & Y \geq \max_j f_j(y) \\ & Y \in R_t \\ & j = 1, 2, \dots, m. \end{aligned} \quad (2)$$

In order to solve (2) (which we do using Powell's new procedure discussed in the next section) we must solve, for each function $f_j(y)$, $j=1,2,\dots,m$, the worst case problem.

$$\begin{aligned} \max \quad & f_j(y) \\ & y \in R_t \end{aligned}$$

Solution of the worst case problem is not easy and is discussed in Section II.

In the variable tolerance case the objective is to find the largest tolerances as well as the nominal design point which results in 100% yield for a feasibility region defined by the constraints

$$f_j(x) \leq c_j \quad j = 1, 2, \dots, m.$$

This problem can be stated as the constrained optimization problem

$$\begin{aligned} & \max y \\ & \text{subject to} \\ & \max f(y) \leq c \\ & y \in V \\ & j = 1, 2, \dots, m \end{aligned} \tag{3}$$

where

$$V = \{y, x_i = \bar{x}_i - y t_i - \bar{x}_i + y t_i\}$$

which is similar to (2) (note $\max y$ is the same as $\min -y$) and can be solved using the same procedure described in Sections II and III.

An example of the new algorithm is given in Section IV.

II. Powell's Method

Consider the problem

$$\min \phi(x) \tag{4}$$

subject to

$$c_i(x) = 0, \quad i = 1, 2, \dots, m$$

where ϕ and c^i are twice differentiable, real valued functions. The Kuhn-Tucker conditions for the solution of (4) are

$$\nabla f(x^*) - \sum_{i=1}^m \lambda_i^* \nabla c_i(x^*) = 0$$

$$c_i(x^*) = 0 \quad i = 1, 2, \dots, m \quad (5)$$

If the set of equations (5) were to be solved using Newton's method we would iteratively solve the equations.

$$\begin{bmatrix} \nabla^2 f(x^k) - \sum_{i=1}^m \lambda_i \nabla^2 c_i(x^k) & -\nabla c_1(x^k) \dots -\nabla c_m(x^k) \\ \nabla c_1^T(x^k) \\ \vdots \\ \nabla c_m^T(x^k) \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \end{bmatrix} = \begin{bmatrix} -\nabla f(x^k) + \sum_{i=1}^m \lambda_i \nabla c_i(x^k) \\ -c_1(x^k) \\ \vdots \\ -c_m(x^k) \end{bmatrix}$$

$$\begin{aligned} x^{k+1} &= x^k + \Delta x^k \\ \lambda^{k+1} &= \lambda^k + \Delta \lambda^k \end{aligned} \quad (6)$$

Powell approximates the second derivative of the Lagrangian

$$L(x, \lambda) = f(x) - \sum_{i=1}^m \lambda_i c_i(x)$$

with a positive definite matrix B that is updated at each step using a BFGS formula, and the step can be smaller than Δx in order to guarantee global convergence [5]. Also instead of solving (6) the equivalent quadratic problem is solved.

$$\begin{aligned} \min \quad & \nabla f^T(x_k)Ax^k + 1/2 Ax^k B Ax^k \\ \text{s.t.} \quad & \begin{bmatrix} \nabla c_1^T(x^k) \\ \dots \\ \nabla c_m^T(x^k) \end{bmatrix} \Delta x^k + \begin{bmatrix} c_1(x^k) \\ \dots \\ c_m(x^k) \end{bmatrix} = 0 \end{aligned} \quad (7)$$

This problem yields the same solution as (6) and the same A's, if B is the exact Hessian of the Lagrangian, and provides an obvious way to handle non-equality constraints, that are just appended to the set of constraints after linearization.

Observe that if near the solution the rank of the Jacobian of the constraints in (6) is n, the last m rows of the matrix can be used to obtain the solution, avoiding the computation of the Hessian. Because we are using Newton's method we still have 2nd order convergence. This fact is why the Madsen and Schjaer-Jacobsen algorithm [1] can, in the regular case, provide a second order rate of convergence without computation of second derivatives [3]. But more important is the observation that we should try to introduce the largest number of possible constraints in order to improve the rate of convergence of the method. This observation is the motivation behind the concept of function splitting which is introduced in the next section.

Note that as with any Quasi-Newton optimization method at least superlinear convergence should be expected [6J].

III. The Worst Case Problem

As was pointed in Section 1, in order to solve (2) we have to solve the worst case subproblera

$$\begin{aligned} \max \quad & f_j(y) \\ y \in & R_t \end{aligned} \tag{8}$$

for each $j = 1, 2, \dots, m$. In general this problem is difficult to solve, and some restrictions need to be made on the f_j . In the following development we will assume that the f_j are such that the solution of (8) is always found at one vertex of R_t . Even with this simplification, we cannot use Powell's method directly because the constraints in (2) are not smooth. A small variation of the centers of the tolerance region can cause a change of the worst case vertex, or the maximum value can be achieved at more than one vertex simultaneously. To solve this problem at each iteration we not only compute the worst case vertex but also introduce new inequalities corresponding to the vertices that have been active in one of the n previous iterations. As was mentioned in Section II, introduction of new constraints can only help the rate of convergence of the algorithm. In the example of Figure 1, where $m = 1$, at the first iteration we would have only one constraint corresponding to vertex V_2 , but in all subsequent iterations we would always have 2 inequalities, for vertex V_2 and V_3 . Geometrically it can be seen that it prevents the algorithm from bouncing around the 45° degree line. In general, any function could be split into $n+1$ constraints, but as will be seen in the examples this situation is very unlikely.

Another way of looking to the algorithm is as follows. Let $R_v(x)$ represent the set of vertices of the tolerance region corresponding to a

nominal value x . Under the assumptions made above, (2) can be recast as

$$\begin{aligned} & \min \gamma \\ \text{s.t. } & \gamma \geq f_j(y_i) & j = 1, \dots, m \\ & & i = 1, \dots, 2^n \\ & & y_i \in R_v(x) \end{aligned} \quad (7)$$

Notice that the number of constraints would be $m \times 2^n$ making the computation prohibitive for large values of n . The proposed method is basically a scheme to pick out those constraints which are good candidates to be active.

IV. Examples

Consider as a first example the fixed tolerance problem (1) with one constraint

$$f(x) = .505x_1^2 + .505x_2^2 - .99x_1x_2$$

with the tolerances $t = (0.1, 0.1)$. The level cuts are elongated ellipses and the minimum of $f(x)$ is obtained at $x = (0., 0.)$ Note that since we have one constraint and two variables (i.e., $m=1$ and $n=2$) this problem is singular even with constraint splitting. Starting with the point $x = (2, 4)$ and $\gamma=1$, the algorithm took 13 iterations to converge to the solution $x = (0, 0)$ requiring 66 function evaluations and 55 gradients,

The variable tolerance problem, with $c = 0.1$ with the same starting point that took 12 iterations to converge to the solution $x = (0, 0)$, $r = 2.236$ requiring 63 functions evaluations and 51 gradients.

When solving the subproblem $\max_{y \in R_t} f_1(y)$ we had a problem whenever

$x_1 = x_2$. The algorithm we use to find y is based upon gradient information. The gradient at the center of R_t is computed and we take the vertex pointed by the signs of this gradient. If the signs of the gradient are the same at this vertex a maximum was found, otherwise we take the vertex pointed by the new gradient and the procedure is repeated. If $x_1 \neq x_2$ it can easily be checked that the maximum found is a local maximum. It was decided that before taking a vertex given by this procedure as the worst case point, we always check the function values at the worst vertices in the last n iterations. Notice that no extra function evaluations are required because the function at these vertices need to be computed anyway due to the splitting mechanism.

For comparison we tried the above algorithm to solve example 1 of [1]- Both fixed tolerance problem and variable tolerance problem took only 5 iterations to converge to the solution. For functions f_2 and f_3 only one vertex becomes active. For function f_1 , 2 vertices are active at the solution.

References

- [1] K. Madsen and H. Schjaer-Jacobsen, "Algorithms for Worst Case Tolerance Optimization." Report NI-78-05, Institute of Numerical Analysis, Technical University of Denmark.
- [2] _____, "Singularities in Minimax Optimization of Networks," IEEE Transactions on CAS, Vol. CAS-23, No. 7, July 1974, pp 456 - 560.
- [3] _____, "Linearly Constrained Minimax Optimization," Mathematical Programming, 14 (1978) 208 - 223,
- [4] M. J. D. Powell, "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," Mathematical Programming 14 (1978) 224 - 248.
- f⁵-! M. J. D. Powell, "A Past Algorithm for Nonlinear Constrained Optimization Calculations", Presented at the 1977 Dundee Conference on Numerical Analysis.
- [6] M. J. D. Powell, "The Convergence of Variable Metric Methods for Nonlinear Constrained Optimization Calculations."

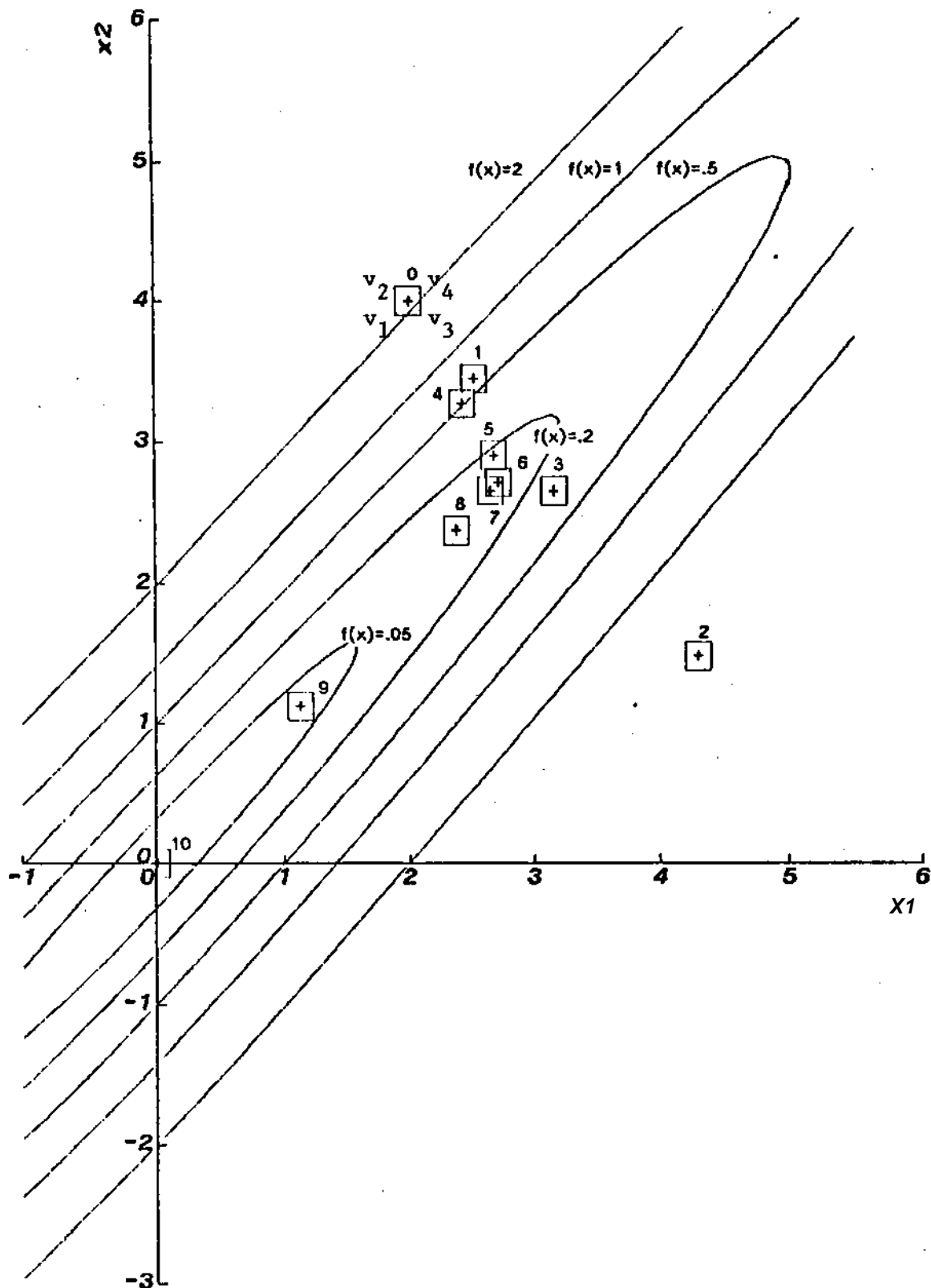


FIG. 1 EXAMPLE 1 ($t=.1, .1$)

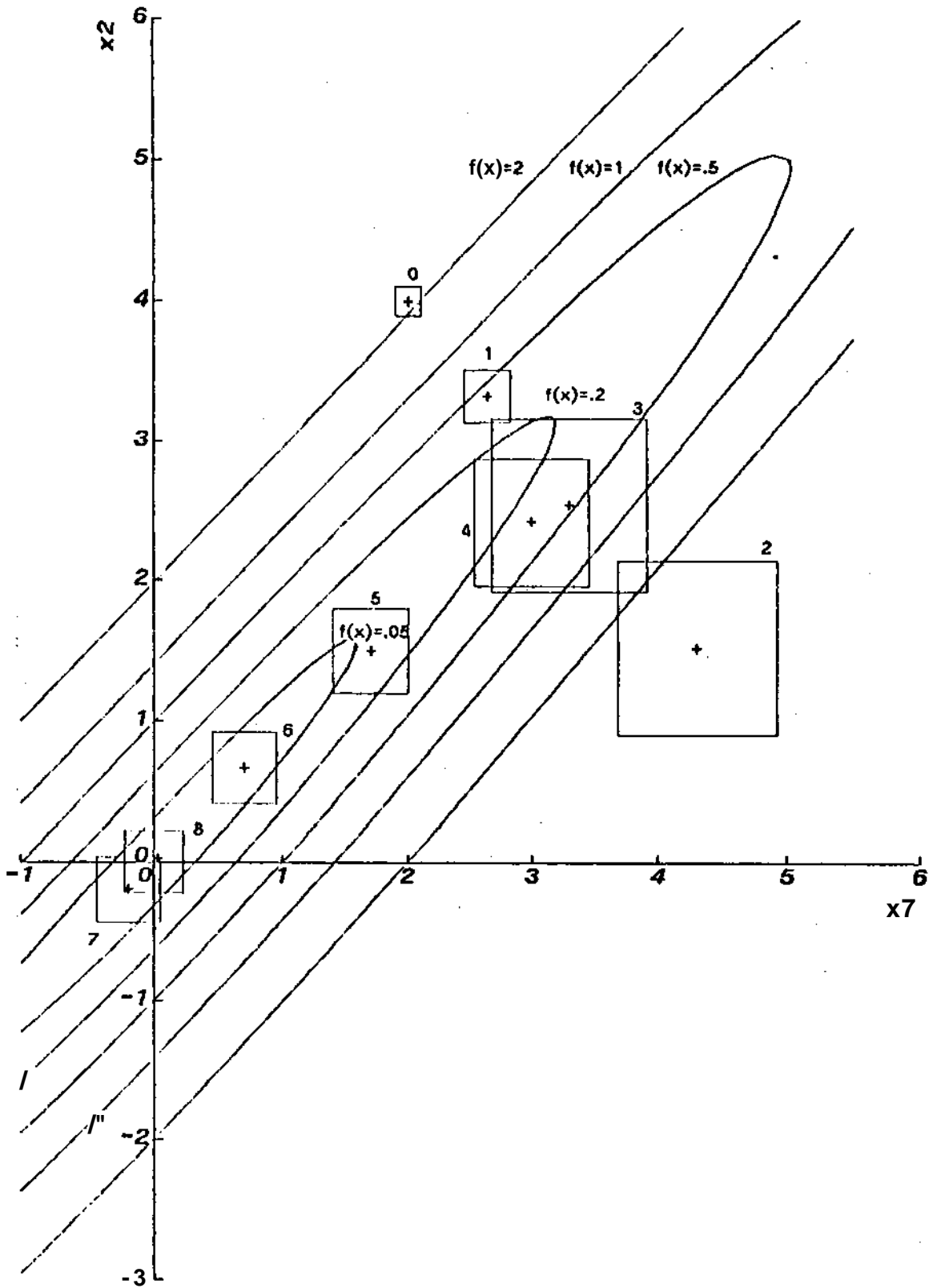


FIG. 2 EXAMPLE 1 ($c=1$)

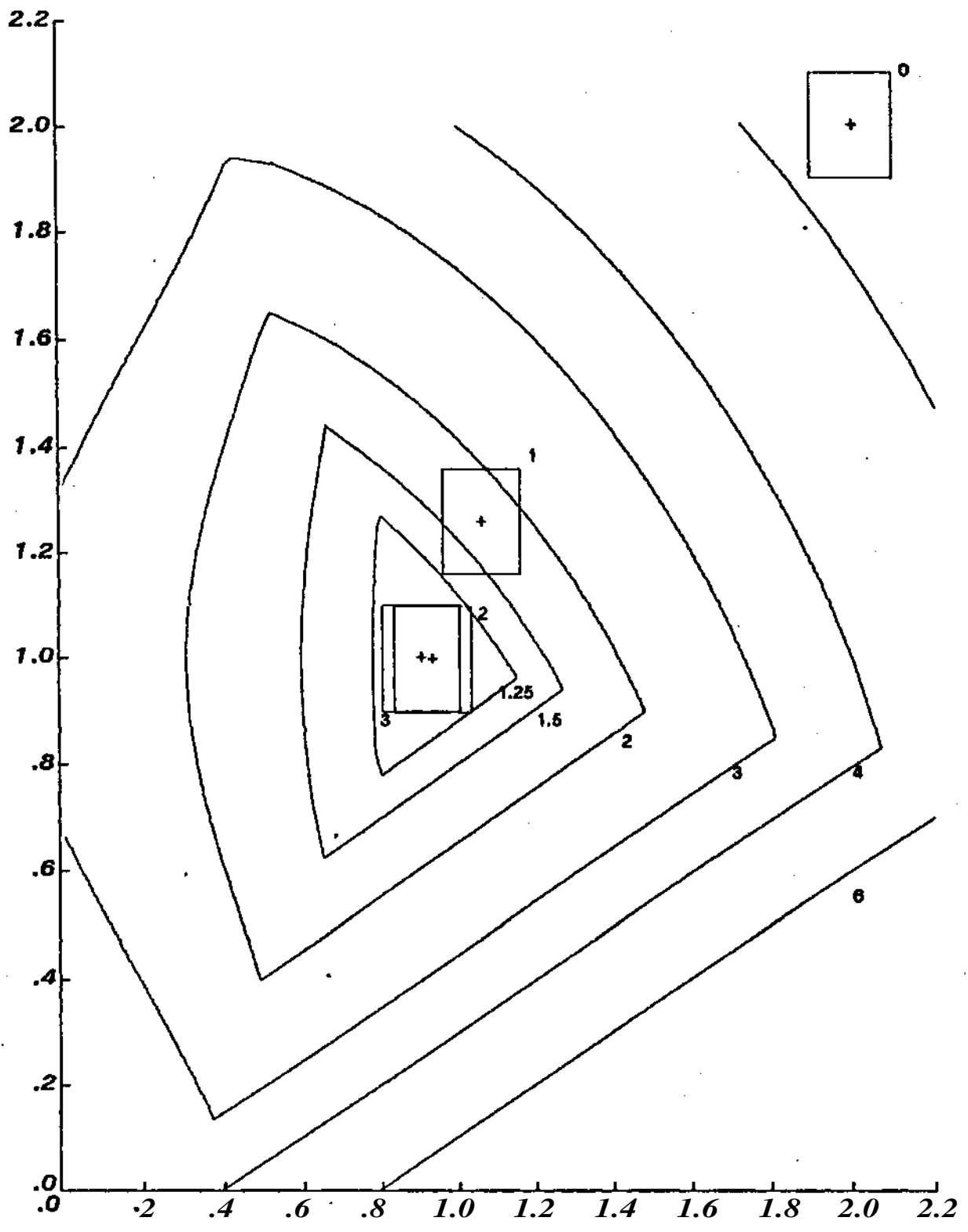


FIG. 3 EXAMPLE 2 (f.1,1)

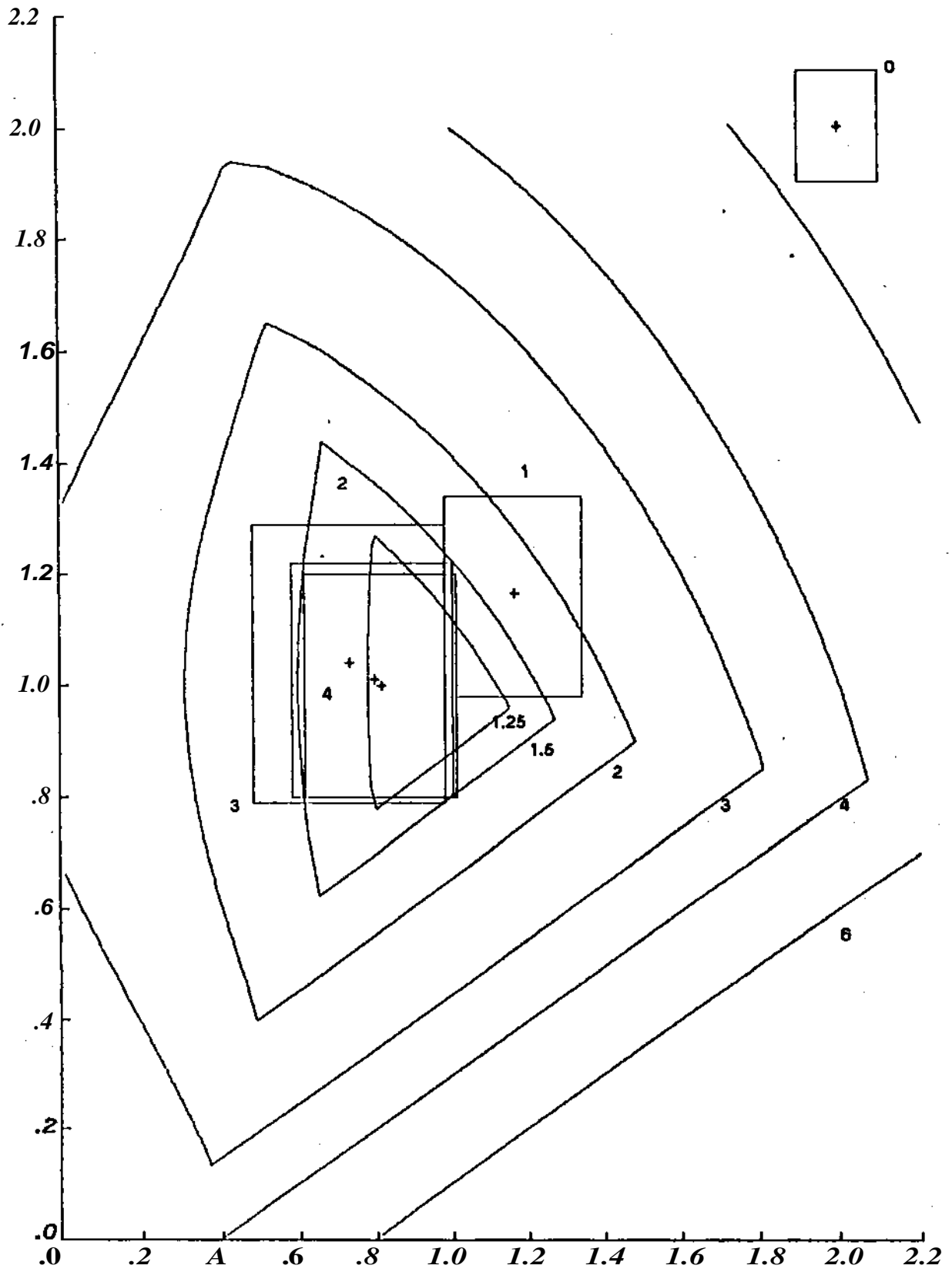


FIG. 4 EXAMPLE 2 ($\tau=1.5$)