Algorithms for worst-case tolerance optimization

Schjær-Jacobsen, Hans; Madsen, Kaj

Published in:
IEEE Transactions on Circuits and Systems

Link to article, DOI:
10.1109/TCS.1979.1084700

Publication date:
1979

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Schjær-Jacobsen, H., & Madsen, K. (1979). Algorithms for worst-case tolerance optimization. *IEEE Transactions on Circuits and Systems*, *26*(9), 775-783. https://doi.org/10.1109/TCS.1979.1084700

Algorithms for Worst-Case Tolerance Optimization

HANS SCHJAER-JACOBSEN, MEMBER, IEEE, AND KAJ MADSEN

Abstract—**New algorithms are presented for the solution of optimum tolerance assignment problems. The problems considered are defined mathematically as a worst-case problem (WCP), a fixed tolerance problem (FTP), and a variable tolerance problem (VTP). The basic optimization problem without tolerances is denoted the zero tolerance problem (ZTP). For solution of the WCP we suggest application of interval arithmetic and also alternative methods. For solution of the FTP an algorithm is suggested which is conceptually similar to algorithms previously developed by the authors for the ZTP. Finally, the VTP is solved by a double-iterative algorithm in which the inner iteration is performed by the FTP- algorithm. The application of the algorithm is demonstrated by means of relatively simple numerical examples. Basic properties, such as convergence properties, are displayed based on the examples.**

I. Introduction

In this work we shall be concerned with methods for the solution of nonlinear minimax optimization problems where the optimization parameters are subject to tolerances. Normally, the solution to classical minimax optimization problems (i.e., no tolerances) may be determined very effectively with an accuracy, which is only limited by the finite word length of the particular computer used for the calculations [1]. Now the situation may well arise, that the actual parameters may be realized only by a much lower accuracy, say, 0.1–10 percent in practical engineering systems. If the optimal solution is sensitive to changes in the parameters, this may completely disturb the optimum and one must think of ways of including the tolerances in the optimization objective.

Other authors have suggested methods for solution of optimum tolerance assignment problems. A popular approach has been to apply general nonlinear programming techniques to minimize a cost function defined in terms of the weighted tolerances and simultaneously impose nonlinear constraints on the worst case. Some examples of research along these lines may be found in [2]–[6]. Alternative approaches have also been reported [7]–[9].

In this paper, like in some of the previous works, we let the n optimization parameters represent the center of an n-dimensional tolerance interval, on which the worst case is calculated. For the solution of this worst-case problem (WCP) we suggest to use interval arithmetic, which guarantees that the true worst case is always found. Otherwise the true worst case may not be found unless certain assumptions—for instance one-dimensional convexity—are satisfied.

Our approach to minimizing the worst case is less general than, for example, the cost function formulation. However, the advantage of our approach is that the worst-case calculations and the minimization are linked together such that each worst-case calculation affects the minimization problem immediately. In the cost function formulation a normal nonlinear problem is set up and a full minimization is carried out before the information given by the worst-case calculations is used to adjust the approximating nonlinear problem. So it seems that when the problem to be solved belongs to the class described below our approach should be used.

One possible formulation is to minimize the worst case by moving the center of the tolerance interval while keeping the tolerance fixed. For this socalled fixed tolerance problem (FTP) we present an algorithm. Another formulation is to maximize the tolerance interval by means of a scaling factor, subject to the constraint, that the minimized worst case does not exceed a given limit. For this variable tolerance problem (VTP) a double iterative algorithm is suggested, which makes use of the fixed tolerance algorithm in the inner iteration. In the next section we shall define the problems more precisely.

II. Mathematical Formulation of the Problems

A. The Zero Tolerance Problem

We consider a set of m nonlinear differentiable functions of n real variables

$$f_j(x) \equiv f_j(x_1, \cdots, x_n), \qquad j = 1, \cdots, m. \tag{1}$$

For easy notation, let us introduce the integer index sets

$$J \equiv \{j \mid j = 1, \cdots, m\}, I \equiv \{i \mid i = 1, \cdots, n\}. \tag{2}$$

The zero tolerance problem (ZTP) is then defined as the classical nonlinear minimax problem of determining

$$F^* \equiv F(x^*) = \min_{x \in R^n} F(x) \tag{3}$$

where

$$F(x) \equiv \max_{j \in J} f_j(x). \tag{4}$$

Manuscript received February 5, 1979; revised April 20, 1979. This work has been supported by the Danish Natural Science Research Council, Grant 511-6874. A major portion of the work was presented at the 1978 IEEE International Symposium on Circuits and Systems, New York, NY.

H. Schjaer-Jacobsen is with A/S LK-NES, Industriparken 20-36, DK-2750 Ballerup, Denmark.

K. Madsen is with the Institute for Numerical Analysis, Technical University of Denmark, DK-2800 Lyngby, Denmark.

0098-4094/79/0900-0775$00.75 ©1979 IEEE

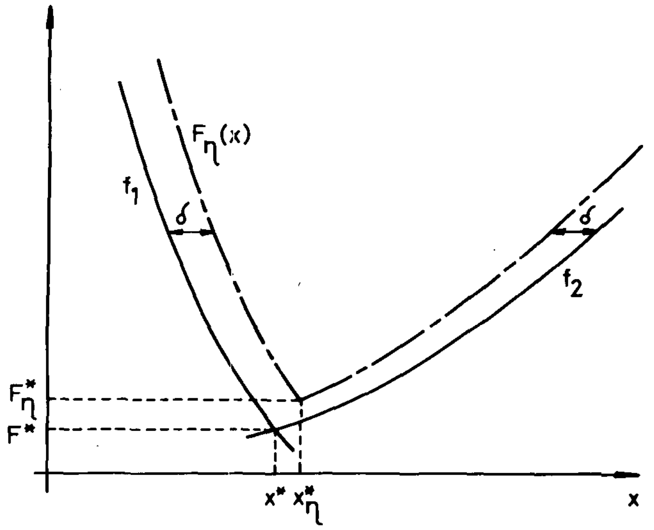

Fig. 1. One-dimensional worst-case objective function $F_\eta(x)$. Solution of FTP indicated by x_η^*, F_η^*.

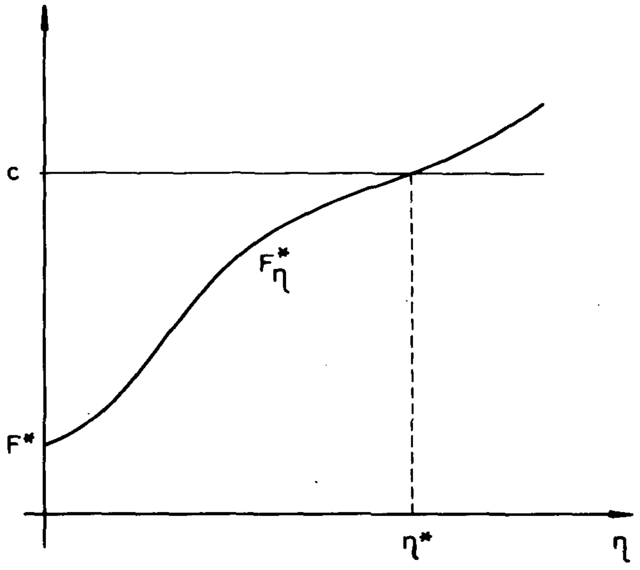

Fig. 2. Illustration of VTP with solution η^*.

B. The Worst-Case Problem

Now, let there be given a vector of tolerances

$$\delta \equiv (\delta_1, \cdots, \delta_n)^T, \qquad \delta_i \geqslant 0 \text{ for } i \in I \tag{5}$$

on the variables vector x. The tolerance interval is defined as

$$\Omega_{x,\eta} \equiv \{ y | (x_i - \eta\delta_i \leqslant y_i \leqslant x_i + \eta\delta_i, \eta \geqslant 0, i \in I)\} \tag{6}$$

and the worst-case objective function is defined as

$$F_\eta(x) \equiv \max_{j \in J, y \in \Omega_{x,\eta}} f_j(y) \tag{7}$$

where

$$y \equiv (y_1, \cdots, y_n)^T. \tag{8}$$

Note that η may be interpreted as a scale factor of the tolerance interval (6). If, for example, $\eta = 1$ the tolerance interval is defined solely in terms of the initially given tolerances (5). Note also that ultimately a global optimization over the tolerance interval (5) is required in order to define a *true* worst-case objective function by (7). This problem we shall identify as the WCP. An example of the worst-case objective function is given in Fig. 1, for $n = 1$ and $m = 2$.

C. The Fixed Tolerance Problem

We now define the fixed tolerance problem (FTP) on the basis of the worst-case objective function (7), namely as that of determining

$$F_\eta^* \equiv F_\eta(x_\eta^*) = \min_{x \in R^n} F_\eta(x) \tag{9}$$

for fixed η. In other words, the location x of the center of the tolerance interval is determined in such a way, that the worst case objective function (7) is minimized. It should be noted, that for η close to zero, we have that x_η^* is close to x^* and consequently F_η^* is close to F^*. The same is true when all components of δ are close to zero. The solution to the FTP is demonstrated in Fig. 1.

D. The Variable Tolerance Problem

We finally define the VTP as that of determining

$$\eta^* \equiv \max \eta \tag{10}$$

subject to the constraint

$$F_\eta^* \leqslant c \tag{11}$$

where the given constant c has to be larger than F^* for obvious reasons. Essentially, we have defined the VTP as a one-dimensional problem of maximizing a simple parameter, namely, η, subject to a nonlinear constraint. An example of the VTP is given in Fig. 2.

III. Description of New Algorithms

A. Solution of the WCP by Interval Arithmetic

In this section we shall suggest the application of interval arithmetic in solution of the WCP. It is assumed that each function $f_j(x)$ can be calculated explicitly on a computer. If, for instance, differential equations have to be solved in order to find $f_j(x)$, then the methods of this section are not applicable.

It is pertinent to introduce some basic concepts from interval arithmetic [10] together with a notation in the context of the problems treated in this report.

An interval function

$$\Phi_j(X) \equiv \Phi_j(X_1, \cdots, X_n) \tag{12}$$

may be derived from f_j by substituting the real variables x by the intervals

$$X \equiv (X_1, \cdots, X_n)^T \tag{13}$$

and by replacing the real arithmetic operations by the corresponding interval arithmetic operations. By definition, f_j is called the *real restriction* of Φ_j. The interval function Φ_j itself is an interval with sure, but often too pessimistic, upper and lower bounds. We are therefore primarily interested in the socalled *interval hull*[1] ϕ_j defined in the following way

$$\phi_j(X) \equiv \{f_j(x) | x \in X\}. \tag{14}$$

It is obvious that

$$\phi_j(X) \subseteq \Phi_j(X). \tag{15}$$

It is generally true, that if the variables appear only once in an interval function, then the interval hull (which gives the narrowest and true interval for the function) is identical to the interval function itself. If the variables appear more than once, then the interval hull may be calculated by iterative methods [11]–[13].

We now return to the solution of the WCP. Consider (7), which may be written

$$F_\eta(x) = \max_{j \in J} \max_{y \in \Omega_{x,\eta}} f_j(y) \tag{16}$$

where the outer maximization is done simply by inspection of the m maxima found by the inner maximization for j attaining integer values from 1 to m.

If each variable x_i appears only once[2] in f_j, the inner maximization is simply carried out by means of (12) since the tolerance interval is an interval vector, which is denoted by Y,

$$Y = \Omega_{x,\eta}. \tag{17}$$

The maximum is then the right endpoint of the resulting interval, which we shall denote by $\tilde{\phi}_j$. For later use the left endpoint is correspondingly denoted by $\tilde{\phi}_j$. Consequently we may write

$$\tilde{\phi}_j(x) = \min_{y \in Y} f_j(y) \qquad \tilde{\phi}_j(x) = \max_{y \in Y} f_j(y) \tag{18}$$

and (16) may be reformulated to

$$F_\eta(x) = \max_{j \in J} \tilde{\phi}_j(x). \tag{19}$$

Although not explicitly written, it is clear that $\tilde{\phi}_j(x)$ and $\tilde{\phi}_j(x)$ are functions of η since they are determined as extrema over the interval Y, which depends on η, see (17) and Fig. 3.

If some variables occur more than once then the method of Skelboe [12] can be used. However, this method is rather time consuming. Instead one could use (12), which still yields a true upper bound, but it might be too pessimistic, especially when the tolerances are large.

It should be explicitly noted here, that by solving the WCP as described above, no information is gained about *where* in the tolerance interval the worst case is attained. As we shall later on make use of the partial derivatives of the functions f_j, the problem arises where to calculate them, and here it turns out that the rate of convergence of the FTP algorithm is influenced positively if the derivatives are calculated in the point where the worst case is attained. We estimate this point by other means, see the following sections.

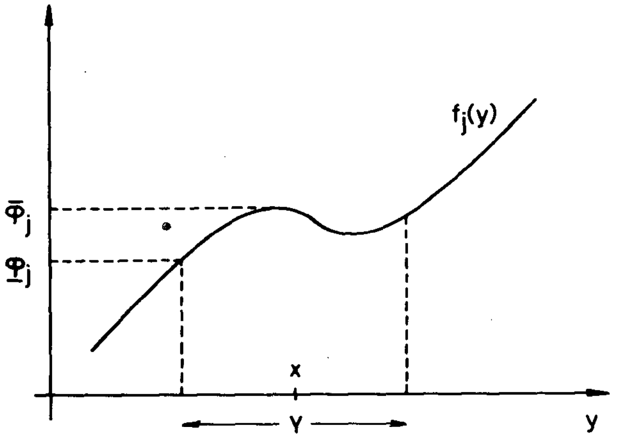

Fig. 3. Extrema of a one-dimensional function over the interval Y.

The functions f_j must necessarily be programmed in interval arithmetic. In order to facilitate this, a Fortran subroutine library has been written, containing the four basic interval operations $+$, $-$, $\cdot$, and $/$. Further, the library contains the most commonly used functions, such as exp, sin, cos, log, etc., programmed in interval arithmetic [14].

B. Solution of the WCP by other Methods

Alternative methods for finding the maximum of the function f_j over the interval $\Omega_{x,\eta}$ have been considered for example in [4] and [6]. These methods rely on the assumption that the function f_j is one dimensionally convex.

We have used a method based on the same principles as those of [4] and [6]. The main idea is that if f_j is monotonic over $\Omega_{x,\eta}$ as a function of each variable x_i, $i \in I$, then the maximum is attained at a vertex of $\Omega_{x,\eta}$, namely at the point

$$y = (y_1, \cdots, y_n)^T \tag{20}$$

where

$$y_i = x_i + \eta \delta_i \cdot \operatorname{sign}(\partial f_j / \partial x_i), \qquad i \in I. \tag{21}$$

In the algorithm we suppose that we have estimates a_{ji} of $\partial f_j / \partial x_i(x)$. These may stem from an earlier evaluation of the derivatives at a point near x. Now the function value and the derivatives are calculated at the point

$$\xi = (\xi_1 \cdots \xi_n)^T \tag{22}$$

for which

$$\xi_i = x_i + \eta \delta_i \cdot \operatorname{sign}(a_{ji}). \tag{23}$$

If

$$\operatorname{sign}(\partial f_j / \partial x_i(\xi)) = \operatorname{sign}(a_{ji}), \qquad i \in I \tag{24}$$

[1] In [10] the term *united extension* is used.

[2] Powers of x_i, x_i^p, count as one occurrence of x_i.

then the Kuhn–Tucker conditions state that ξ is a local maximum over $\Omega_{x,\eta}$. If some of the signs are not identical, then we let $a_{ji} = \partial f_j / \partial x_i(\xi)$ and find a new vertex using (23). If (24) holds we are finished, otherwise we go through a new cycle. This continues as long as the number of equalities in (24) increases. If (24) is not satisfied for all $i \in I$ when we stop, the largest value of $f_j(\xi)$ found during the cycle is used as the worst case. In one special situation, however, we finish in another way. If the number of correct signs in (24) is $(n-1)$ in two consecutive vertices and the wrong sign is attained for identical values of i, then a point between the two vertices is found by linear interpolation, and the search is finished.

Of course this search for the worst case could be further developed, but this is beyond the scope of the present paper.

C. An Algorithm for the FTP

We consider the problem of minimizing the worst-case objective function $F_\eta(x)$ defined by (7) and reformulated in (19). Accordingly, in this section, we write the worst-case objective function

$$F_\eta(x) = \max_{j \in J} \tilde{\phi}_j(x) \tag{25}$$

where $\tilde{\phi}_j$ is explicitly written as a function of x, the center of the tolerance box. The resemblance of (25) with the classical zero tolerance minimax problem is obvious. It is therefore suggested to develop an algorithm for minimization of (25) which is similar to [15]. A flow diagram is shown in Fig. 4.

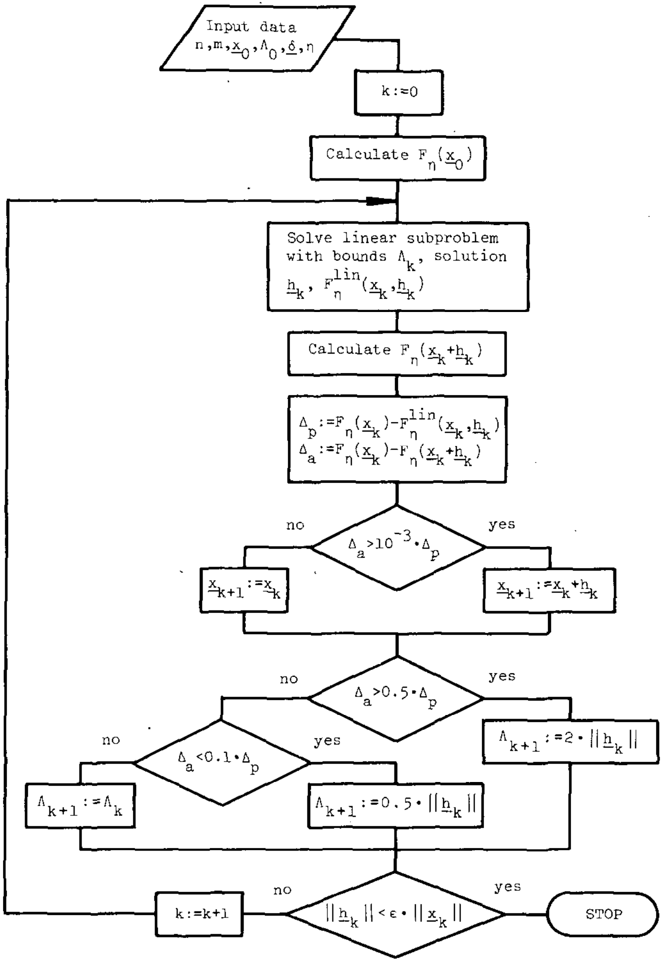

Fig. 4. Simplified flow diagram of the FTP algorithm.

Suppose that an approximation x_k to x_η^* is known at the kth step of the algorithm. Then a better approximation x_{k+1} is sought for and to this end we use the fact that

$$\tilde{\phi}_j(x_k + h) \cong \tilde{\phi}_j(x_k) + \tilde{\phi}_j'(x_k)^T \cdot h, \qquad j \in J \tag{26}$$

for a sufficiently small value of $\|h\|_\infty$. By $\tilde{\phi}_j'$ we denote the vector of partial derivatives

$$\tilde{\phi}_j'(\cdot) = \left(\frac{\partial \tilde{\phi}_j}{\partial x_1}(\cdot), \cdots, \frac{\partial \tilde{\phi}_j}{\partial x_n}(\cdot) \right)^T, \qquad j \in J. \tag{27}$$

Notice, that these derivatives may be calculated by the WCP algorithm since $\partial \tilde{\phi}_j / \partial x_i(x) = \partial f_j / \partial x_i(y)$, y being estimated in the WCP algorithm. The approximation (26) suggests a linear minimax problem in h. Minimize

$$F_\eta^{\text{lin}}(x_k, h) = \max_{j \in J} \left(\tilde{\phi}_j(x_k) + \tilde{\phi}_j'(x_k)^T \cdot h \right) \tag{28}$$

subject to the constraint

$$\|h\|_\infty \leqslant \Lambda_k \tag{29}$$

where $\Lambda_k > 0$ is a bound which is adjusted in each k step of the algorithm according to the quality of the approximation (26).

Let the solution to the linear subproblem be denoted by h_k. The quantities Δ_a and Δ_p are defined as the actual decrease in F_η and the decrease predicted by the linear approximation, respectively,

$$\Delta_a = F_\eta(x_k) - F_\eta(x_k + h_k) \tag{30}$$

and

$$\Delta_p = F_\eta(x_k) - F_\eta^{\text{lin}}(x_k, h_k). \tag{31}$$

In order to accept $x_{k+1} = x_k + h_k$ as the next iterate, it is tested if Δ_a exceeds a small multiple of Δ_p,

$$\Delta_a \geqslant 10^{-3} \Delta_p. \tag{32}$$

Otherwise $x_{k+1} = x_k$.

The strategy for adjusting the bounds Λ_k is also based on comparison between the quantities Δ_a and Δ_p. If

$$\Delta_a \geqslant 0.5 \Delta_p \tag{33}$$

the linear approximations seem to be very good, and consequently we allow for a larger step length

$$\Lambda_{k+1} = 2\|h_k\|_\infty. \tag{34}$$

If, on the other hand,

$$\Delta_a \leqslant 0.1 \Delta_p \tag{35}$$

then the linear predictions are too wide from the actual

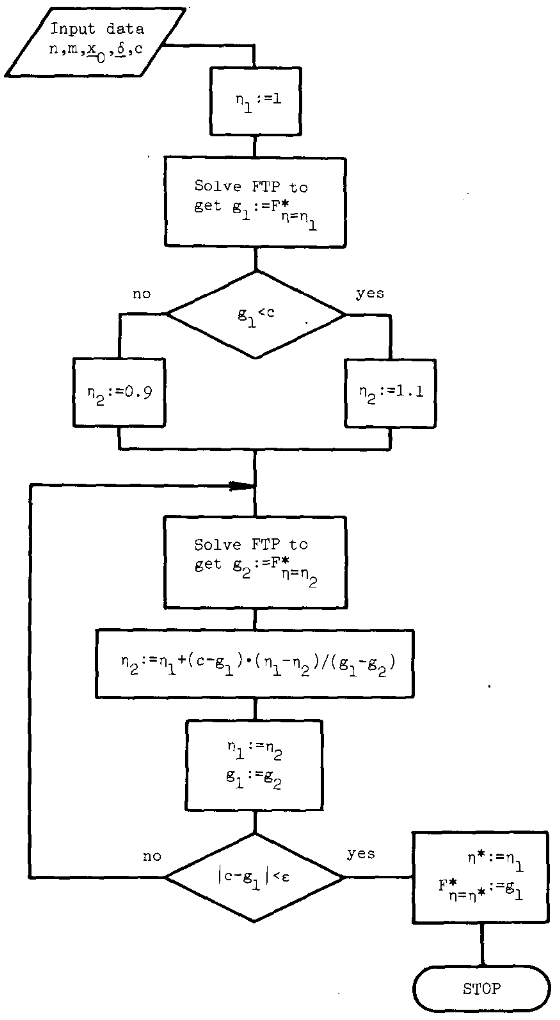

Fig. 5. Simplified flow diagram of the VTP algorithm.

decrease and the step length is shortened:

$$\Lambda_{k+1} = 0.5\|\boldsymbol{h}_k\|_\infty. \tag{36}$$

In all other cases the step length is considered adequate,

$$\Lambda_{k+1} = \|\boldsymbol{h}_k\|_\infty. \tag{37}$$

The choice of constants in (32)–(36) is based on numerical experiments with the algorithm.

The criterion used for convergence of the iterations is

$$\|\boldsymbol{h}_k\|_\infty \leqslant \epsilon \|\boldsymbol{x}_k\|_\infty \tag{38}$$

where ϵ is a small positive number specified by the user. Also an upper bound for the number of iterations may be specified by the user. The iterations are also stopped when $\Delta_p \leqslant 0$, indicating dominating rounding errors in the solution of the linear problem.

D. A Double-Iterative Algorithm for the VTP

The algorithm presented in this section for the VTP is based on repeated application of the previously described algorithm for the FTP. The problem is to determine the intersection between the function F^* and the constant c (see (10), (11), and Fig. 2). This is done by a Regula-Falsi method where each iteration involves one solution of the FTP. In Fig. 5 is shown a simplified flow diagram of the algorithm. We thus simultaneously determine the maximum value of η called η^*, and the center of the tolerance interval $\boldsymbol{x} = \boldsymbol{x}^*_{\eta=\eta^*}$ under the condition that the constraint (11) is satisfied.

It is of course essential that the inner iteration, namely solution of the FTP, is very fast. Therefore, the quadratic convergence of the FTP algorithm obtained in "normal" cases is of central importance.

If the solution F^* of the zero tolerance problem is not known when the VTP iterations are started, it might happen that c in (11) is specified such that $c < F^*$, which means that the set of feasible points is empty. We have built in a check in the algorithm to detect such a failure, usually resulting in a negative value of η.

IV. Numerical Examples

The following examples have all been calculated in double precision Fortran IV (approximately 16 decimal digits) on an IBM 370/165 computer. The computer programs are described in [14].

A. Example 1

In this example we consider a set of three functions in two variables

$$\begin{aligned} f_1(x_1,x_2) &= \exp(-x_1+1)\cdot((x_2-1)^2+1) \\ f_2(x_1,x_2) &= \exp(x_1-2x_2+1) \\ f_3(x_1,x_2) &= x_1^2+x_2^2-1. \end{aligned} \tag{39}$$

A contour plot of the zero tolerance objective function $F(\boldsymbol{x})$ is shown in Fig. 6. We have that $F^*=1$ for $\boldsymbol{x}^* = (1,1)^T$.

First the fixed tolerance problem with $\boldsymbol{\delta}=(0.1,\ 0.1)^T$ and $\eta=1$ is solved using the starting values $\boldsymbol{x}_0=(2,2)^T$ and $\Lambda_0 = 0.1$ and $\epsilon = 10^{-4}$. The solution $\boldsymbol{x}^*_\eta = (0.9064737742251549,\ 1.00136277924813)^T$ was found after 8 iterations.[3] The worst-case values for the three functions were identical within the computer accuracy: $\tilde{\phi}_1(\boldsymbol{x}^*_\eta) = \tilde{\phi}_2(\boldsymbol{x}^*_\eta) = \tilde{\phi}_3(\boldsymbol{x}^*_\eta) = 1.22598942976934$. This high accuracy obtained in very few iterations indicates a quadratic convergence rate of the FTP algorithm.

Until now it has not been specified whether the above results were obtained by using interval arithmetic to solve the WCP or the alternative algorithm of Section III-B. Actually the two methods behave exactly identical for this problem, the reason being that the functions (39) are monotone over $\Omega_{x,\eta}$ for $\boldsymbol{x}$ near $\boldsymbol{x}^*$ and therefore the worst cases are all attained at the vertices of the tolerance interval and are equally well found by the two methods.

We next consider the variable tolerance problem defined by the above parameters and the value of $c=1.5$. Again the two alternative WCP methods find the same

[3] One iteration is defined as one cycle of the FTP algorithm. In other words, one iteration corresponds to solving the linearized subproblem once or to one solution of the WCP.

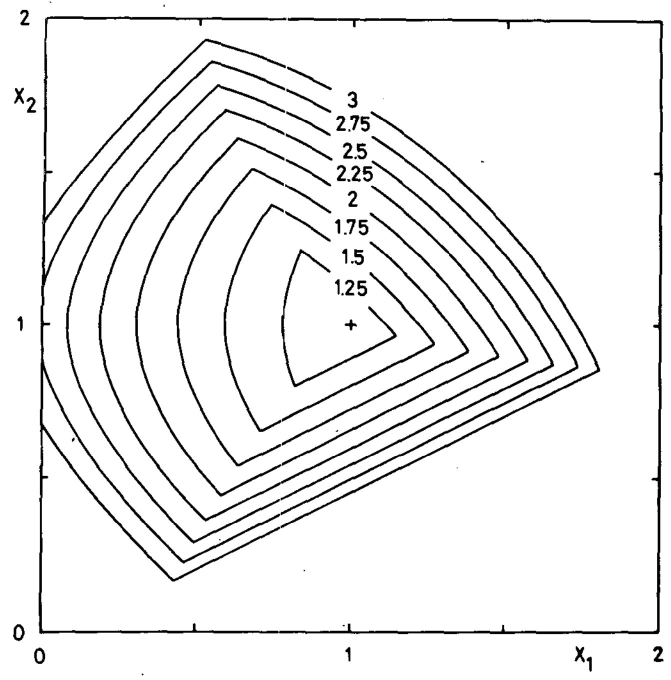

Fig. 6. Contour plot of zero tolerance objective function from Example 1.

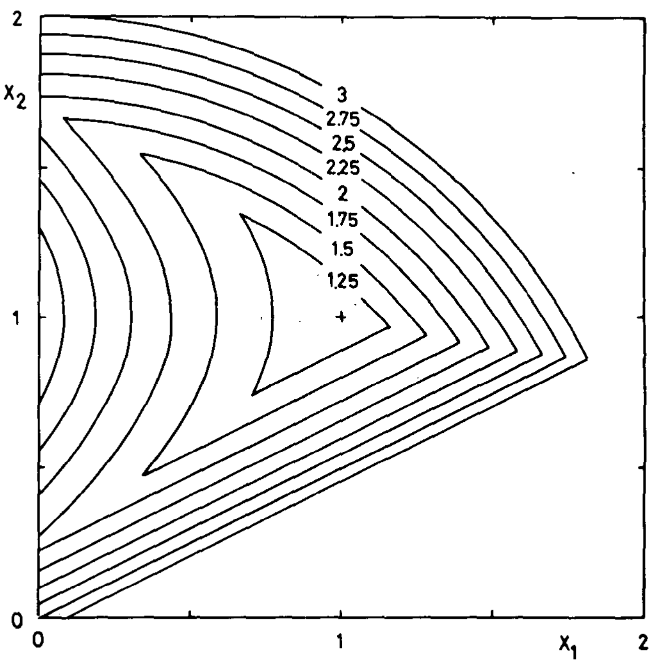

Fig. 7. Contour plot of zero tolerance objective function from Example 2.

solution using equally many iterations. The solution found is $\underset{\sim}{x}_{\eta}^{*}=(0.830691342600859,\ 1.00654359705450)^{T}$ and $\tilde{\phi}_1=\tilde{\phi}_2=\tilde{\phi}_3=1.49995510078898$ which took 21 iterations. The tolerances have been improved from $\eta\delta_1=\eta\delta_2=0.1$ to $\eta^*\delta_1=\eta^*\delta_2=0.195943675447343$.

For practical purposes the high accuracy presented in this example would be more than sufficient, considering the nature of the problem. It is important, however, to study the performance of the algorithm in order to gain insight into the convergence properties. For example, the value of $\epsilon=10^{-4}$ actually used seems to produce values of the worst cases which are correct to 5 digits (they should have been 1.5). With $\epsilon=10^{-6}$ we get 1.50000005 using 26 iterations and with $\epsilon=10^{-8}$ we get 1.499999999998 in 32 iterations.

B. Example 2

This example is identical to the foregoing example, except that

$$f_1(x_1,x_2)=\exp(-x_1+1)/\left((x_2-1)^2+1\right) \tag{40}$$

resulting in a concave zero tolerance function $F(x)$, see Fig. 7. Again $F^*=1$ for $x^*=(1,1)^T$.

In this example we expect the interval arithmetic calculation of the worst case to be different from the alternative method not using interval arithmetic. Consider first the FTP as defined in the preceding example. We get the solutions shown in Table I, where the worst case is shown with as many digits as were identical in the three individual worst cases.

TABLE I
SOLUTION OF FTP OF EXAMPLE 2, $\epsilon=10^{-4}$

	Solution		Worst Case	No. of ite- rations
Interval Arithmetic	0.902102207	1.00210214	1.218837	8
Alternative Method	0.902094885	1.00210338	1.218826	8

The worst case obtained by using the interval arithmetic approach is "worse" than by using the alternative approach. However, the former is the *true* worst case. Since the number of iterations is the same as for Example 1, one might think that the speed of convergence is the same for the two examples. This is not so. The accuracy obtained in Example 1 in 8 iterations is much better (namely the full machine accuracy) than in Example 2. The reason for the poorer accuracy in Example 2, is that the partial derivatives of f_1 are not calculated exactly in the point where the worst case is attained, but rather in a point estimated by linear interpolation. In order to achieve the full machine accuracy in the present example, 12 iterations were necessary. (A much smaller value of ϵ has to be specified, namely $\epsilon=10^{-12}$.) This is still a rather good speed of convergence.

Now consider the variable tolerance problem corresponding to the one solved in Example 1, i.e., $c=1.5$. The results of the double-iterative algorithm are shown in Table II, where the third column contains the optimized tolerance, i.e., the value $\eta^*\delta_1=\eta^*\delta_2$. Again, it is characteristic, that the interval method has detected the true worst case whereas the alternative method has only detected a less strict approximation to it. This may be seen from the larger tolerance obtained by the alternative method and

TABLE II
SOLUTION OF VTP OF EXAMPLE 2, $\epsilon = 10^{-4}$

	Solution		Worst Cases	Tolerances $\eta^*\delta_1=\eta^*\delta_2$	No. of iterations
Interval Arithmetic	0.80201121	1.0094752	1.4999821 1.4999833 1.4999833	0.20746	22
Alternative Method	0.80179387	1.0094967	1.4999830 1.4999838 1.4999838	0.20755	22

the smaller value of the first component of the solution, compare with the concave part of the objective function in Fig. 7.

C. Example 3

The purpose of this example is twofold. First, it is intended to demonstrate the application of the algorithms where, instead of (7), the worst case objective function is defined by

$$F_\eta(x) = \max_{j\in J,\, y\in\Omega_{x,\eta}} |f_j(y)|. \tag{41}$$

The interval arithmetic is particularly well suited for this type of objective function, since by calculating the interval hull we get at the same time $\tilde{\phi}_j$ as well as $\underset{\sim}{\phi}_j$, see (18). Then (41) may be written as

$$F_\eta(x) = \max_{j\in J}\left(-\underset{\sim}{\phi}_j, \tilde{\phi}_j\right) \tag{42}$$

and a linear subproblem corresponding to (28) and (29), but with twice as many functions, may be formulated.

Secondly, the example demonstrates that the FTP algorithm may converge slowly, when it is applied to singular problems. This term is defined in analogy to a singularity in the zero tolerance case [16].

Consider the three functions in two variables

$$f_1(x_1,x_2) = 1.5 - x_1(1-x_2)$$
$$f_2(x_1,x_2) = 2.25 - x_1(1-x_2^2)$$
$$f_3(x_1,x_2) = 2.625 - x_1(1-x_2^3). \tag{43}$$

The zero tolerance function

$$F(x) = \max_{j\in J} |f_j(x)| \tag{44}$$

is shown in Fig. 8. The solution is $F^* = F(x^*) = 0$, $x^* = (3.0, 0.5)^T$.

First the FTP is solved with $\delta = (0.1, 0.1)^T$, $\eta = 1$, $x_0 = x^*$, $\Lambda_0 = 0.1$, and varying ϵ. Using interval arithmetic for the WCP the results in Table III are obtained, from which the following observations may be made.

First, the FTP is singular because the solution is determined by only two of the worst cases (indicated underlined in the table) namely $\tilde{\phi}_2$ and $-\underset{\sim}{\phi}_1$, respectively. The singularity causes a slow convergence, compare, for example, with Table I.

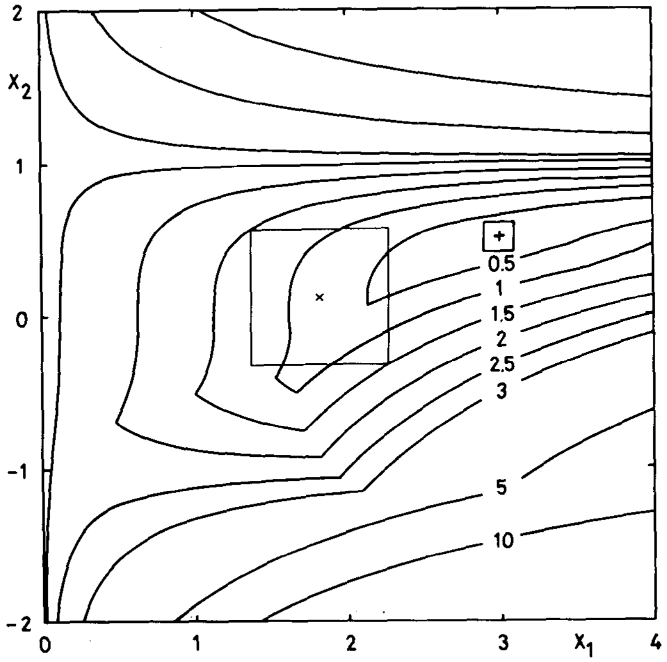

Fig. 8. Contour plot of zero tolerance objective function from Example 3. Initial and optimum tolerance intervals.

TABLE III
SOLUTION OF FTP OF EXAMPLE 3, USING INTERVAL ARITHMETIC

ϵ	Solution $\underline{x}^*_\eta$		j	Φ_j and Ψ_j at Solution		No. of Iterations
10^{-2}	2.880	0.470	1 2 3 1 2 3	0.3066 $\underline{0.3754}$ 0.3617 $\underline{0.3752}$ 0.3205 0.2033	Φ_j $-\Psi_j$	10
10^{-4}	2.89520	0.473879	1 2 3 1 2 3	0.30890488 $\underline{0.37536026}$ 0.35808827 $\underline{0.37536024}$ 0.32651829 0.21366665		34
10^{-6}	2.895250	0.4738887	1 2 3 1 2 3	0.308912143604 $\underline{0.375360255898}$ 0.358077536636 $\underline{0.375360255894}$ 0.326536395275 0.213698122281		52
10^{-8}	2.89525213	0.473889018	1 2 3 1 2 3	0.3089123676317356 $\underline{0.3753602558962728}$ 0.3580772057155057 $\underline{0.3753602558962728}$ 0.3265368604383558 0.2136990925808919		74

Second, because the problem has been solved with four decreasing values of ϵ, it may be observed, that when the worst cases are identical to p digits, then the solution vector is only known with approximately $p/2$ digits. This phenomenon is characteristic for singular minimax problems and has been experimentally verified in many other cases.

The VTP with the same starting conditions as the FTP and $c = 1.5$ is subsequently solved. The results are shown in Table IV for the accuracy specification $\epsilon = 10^{-4}$. As

TABLE IV
SOLUTION OF VTP OF EXAMPLE 3, USING INTERVAL ARITHMETIC, $\epsilon = 10^{-4}$

Solution $\underline{x}^*_{\eta^*}$		j	ϕ_j and $-\psi_j$ at Solution	η^*	No. of Iterations
1.8417	0.13374	1	0.91241834	4.4543	171
		2	1.32210300		
		3	$\underline{1.49999927}$		
		1	$\underline{1.49999925}$		
		2	0.03713868		
		3	-0.26860987		

expected the number of iterations is rather large due to the singularity of the FTP as previously described. It is seen that the worst case is determined by $\tilde{\phi}_3$ and $-\phi_1$ and that the final optimized tolerances are $\eta^*\delta_1 = \tilde{\eta}^*\delta_2 = 0.44543$.

By solving the problems in this example using the alternative method instead of the interval method, one gets the same behavior of the algorithm because the functions are monotonic in the region of interest.

D. Example 4

Let it be desired to approximate the function e^y in the interval $-1 \leqslant y \leqslant 1$ by minimizing the objective function

$$F(x) = \max_{1<j<21} |f_j(x)| \tag{45}$$

where

$$f_j(x_1, \cdots, x_5) = \frac{x_1 + x_2 y_j}{1 + x_3 y_j + x_4 y_j^2 + x_5 y_j^3} - e^{y_j}$$

$$y_j = -1 + 0.1(j-1), \qquad j = 1, \cdots, 21. \tag{46}$$

The solution to this ZTP is known to be

$$x^* = (0.999879, 0.253588, -0.746608, 0.245202, -0.037490)^T$$

$$F^* = 0.122 \cdot 10^{-3}. \tag{47}$$

Now, let there be given a vector of tolerances, which are approximately 1 percent of the (absolute) value of the components in x^*

$$\delta = (10^{-2}, 2.5 \cdot 10^{-3}, 7.5 \cdot 10^{-3}, 2.5 \cdot 10^{-3}, 3.5 \cdot 10^{-4})^T. \tag{48}$$

For different values of η we then solve the FTP allowing us to display F^*_η as a function of η in Fig. 9. From this figure it becomes evident that for small tolerances on the parameters, 10^{-4}–10^{-3} percent, say, the quality of the approximation is rather unaffected. For larger tolerances, however, the approximation is rather sensitive and the value of F^*_η is rapidly increasing for increasing tolerances. For $\eta = 1$, corresponding to 1-percent tolerances (not shown in the figure), the optimum is completely destroyed with the value of $F^*_\eta = 6 \cdot 10^{-2}$.

Typically, in this example, one solution of the FTP required 10 iterations using $x_0 = (0,0,0,0,0.5)^T$, $\Lambda_0 = 0.1$, and $\epsilon = 10^{-6}$.

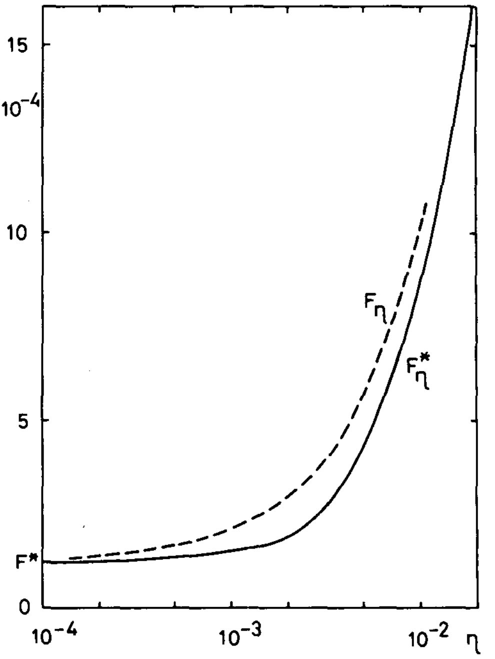

Fig. 9. Solution of fixed tolerance problem of Example 4 as a function of η.

V. CONCLUSION

Based on the algorithms and numerical results presented the following main conclusions may be drawn. By breaking down the worst-case optimization problem into smaller, mathematically well defined subproblems, it has been possible to develop rather effective algorithms for the solution of these subproblems, and to put the different algorithms together in such a way that the full problem may be solved. The advantages of using the interval arithmetic approach in the solution of the WCP are obvious. The worst case is calculated exactly, even for nonconvex functions. However, the method has to be further developed in order to be able to handle the worst case problem satisfactory in general. When the interval arithmetic approach is not feasible, the alternative worst case approach in Section III-B may be applied. In many cases of practical interest this method gives the correct answer.

The fixed tolerance problem is solved by the proposed algorithm with a final rate of convergence which is conjectured normally to be quadratic. This algorithm may be further developed so that cases where one of the nonlinear functions determines more than one worst case may also be treated. The variable tolerance problem is solved by repeated application of the fixed tolerance algorithm.

A relatively large number of numerical examples have been tried by the algorithms. We think that the results presented look rather promising and we suggest that some of the ideas be subject of further research.

REFERENCES

[1] J. Hald and K. Madsen, "A 2-stage algorithm for minimax optimization," Institute for Numerical Analysis, Technical University of Denmark, Lyngby, Denmark, Rep. NI-78-11, Sept. 1978.

[2] J. F. Pinel and K. A. Roberts, "Tolerance assignment in linear networks using nonlinear programming," *IEEE Trans. Circuit Theory*, vol. CT-19, pp. 475–479, Sept. 1972.

[3] J. W. Bandler, "Optimization of design tolerances using nonlinear programming," *J. Optimization Theory and Appl.*, vol. 14, pp. 99–114, July 1974.

[4] J. W. Bandler, P. C. Liu, and J. H. K. Chen, "Worst case network tolerance optimization," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-23, pp. 630–641, Aug. 1975.

[5] J. W. Bandler, P. C. Liu, and H. Tromp, "A nonlinear programming approach to optimal design centering, tolerancing, and tuning," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 155–165, Mar. 1976.

[6] H. Tromp, "The generalized tolerance problem and worst case search," *Proc. IEEE Conf. Computer-Aided-Design of Electronic and Microwave Circuits and Systems* (Hull, England), pp. 72–77, July 1977.

[7] E. M. Butler, "Realistic design using large-change sensitivities and performance contours," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 58–66, Jan. 1971.

[8] S. W. Director and G. D. Hachtel, "The simplicial approximation to design centering," *IEEE Trans. Circuits Syst.*, vol. CAS-24, pp. 363–372, July 1977.

[9] K. S. Tahim and R. Spence, "Statistical circuit analysis—A practical algorithm for linear circuits," *Proc. of the 1978 IEEE Int. Symp. Circuits Syst.* (New York), pp. 180–184, May 1978.

[10] R. E. Moore, *Interval Analysis.*, Englewood Cliffs, NJ, Prentice-Hall 1966.

[11] S. Skelboe, "Computation of rational interval functions," *BIT*, vol. 14, pp. 87–95, 1974.

[12] ——, "True worst-case analysis of linear electrical circuits by interval arithmetic," *IEEE Trans. Circuits Syst.*, to be published.

[13] R. E. Moore, "On computing the range of a rational function of n variables over a bounded region," *Computing*, vol. 16, pp. 1–15, 1976.

[14] K. Madsen and H. Schjaer-Jacobsen, "Fortran program package for worst case tolerance optimization," Inst. for Numerical Analysis, Tech. Univ. of Denmark, Rep. NI-78-09, July 1978.

[15] K. Madsen, H. Schjaer-Jacobsen, and J. Voldby, "Automated minimax design of networks," *IEEE Trans. Circuits Syst.*, vol. CAS-22, pp. 791–796, Oct. 1975.

[16] K. Madsen and H. Schjaer-Jacobsen, "Singularities in minimax optimization of networks," *IEEE Trans. Circuits Syst.*, vol. CAS-23, pp. 456–460, July 1976.

Hans Schjaer-Jacobsen (S'70–M'71) was born in Løgumkloster, Denmark, in 1943. Upon being educated as an electronics technician, he received the Civilingeniør (M.Sc.) and Lic. Techn. (Ph.D.) degrees, both in electrical engineering, from the Technical University of Denmark (TUD), in 1972 and 1975, respectively.

From 1972 to 1975 he was with the Electromagnetics Institute, TUD. He spent the academic year 1975 to 1976 with the Institut für Hochfrequenztechnik, Technische Universität, Braunschweig, Federal Republic of Germany, holding an Alexander von Humboldt research fellowship. From 1976 to 1978 he was with the Institute for Numerical Analysis, TUD. He is now with the company A/S LK-NES, Ballerup, Denmark.

Kaj Madsen was born in Denmark in 1943. He received the Cand. Scient. degree in mathematics from the University of Aarhus in 1968.

He has been a lecturer in numerical analysis since 1968, apart from the academic year 1973 to 1974, when he was at AERE Harwell, Didcot, England. During the summer of 1978 he visited McMaster University, Hamilton, Canada. He is participant of an (IIASA-sponsored) international working group on nondifferentiable optimization. His fields of interest in teaching and research are optimization and interval analysis.