WORKING PAPER
ALFRED P. SLOAN SCHOOL OF MANAGEMENT

A COMPARATIVE STUDY OF SOLUTIONS TO THE HOLT, MODIGLIANI, MUTH AND SIMON DISAGGREGATION MODEL BY SEARCH TECHNIQUES

DAN I. CANDEA

WP 814-75 October 1975

MASSACHUSETTS
INSTITUTE OF TECHNOLOGY
50 MEMORIAL DRIVE
CAMBRIDGE, MASSACHUSETTS 02139

A COMPARATIVE STUDY OF SOLUTIONS TO THE HOLT, MODIGLIANI, MUTH AND SIMON DISAGGREGATION MODEL BY SEARCH TECHNIQUES

DAN I. CANDEA

WP 814-75 October 1975

HD28
.M414
no. 814-75

ACKNOWLEDGMENT

The study resulting in this publication was made under a fellowship granted by the Ford Foundation. However, the conclusions, opinions, and other statements in this publication are those of the author and are not necessarily those of the Ford Foundation.

A B S T R A C T

The paper solves the HMMS disaggregation model using numerical methods. The first proposed approach is to turn the original problem into an unconstrained minimization and apply lattice-Fibonacci search. Then the model's solution is considered as a root-finding problem and two approaches are compared: Bolzano's method, and a modified lattice-Fibonacci technique proposed in the paper. These search algorithms proved extremely efficient in yielding solution estimates with errors of the order of tenths of a percent and less in a small number of iterations.

C O N T E N T S

1. INTRODUCTION

The issue of making aggregate decisions /1/ in capacity planning problems is a direct consequence of the uncertain environmen in which production takes place and of our limited ability in gathering and processing very large amounts of detailed data. Moreover, a manager's approach to the capacity planning question is by its nature aggregate in the first place, in order to give him a broad view of where he stands. Then, naturally, the issue of disaggregating the information generated by the aggregate analysis stage is the next question, which is in no way simpler than the first one /2/.

The book of Holt, Modigliani, Muth and Simon /3/ is a self-contained work that addresses both decision processes pointed out above. By using a quadratic inventory cost function together with the overtime, idle time, hiring, layoff, and other related cost functions (all quadratic), the authors obtain linear decision rules for making optimal aggregate decisions for each period. The aggregation is an extreme one of all product types into a single category requiring, of course, the use of appropriate compatible units that allow the transformation to be made. The linear decision rules specify the aggregate production and work force for each period. Then, using the aggregate decision as a constraint on the other more detailed and numerous decisions that concern the production of individual items, rules are obtained to yield the optimal decisions under the constraint.

There are 4 disaggregation models presented in the

HMMS book:

1- determining production quantities by minimizing inventory holding costs and set-up costs subject to the aggregate inventory constraint (/3/,ch. 10);

2- determining buffer stocks by trading off inventory costs for stock-out costs under the aggregate constraint and disregarding the batch characteristic of the production (/3/, ch.11);

3- planning order points given fixed lot sizes, by minimizing inventory holding costs and inventory depletion costs under aggregate constraint (/3/, ch.12);

4- determining lot sizes and safety stocks by minimizing setup costs, inventory holding and depletion costs, under aggregate constraint (/3/, ch.13).

Models 2, 3, and 4 above require the estimation of the cost of being out of stock by one unit in order to compute the inventory depletion costs. From the managerial point of view this is a difficult task, hard to implement because the intangibles involved in stock-outs can not always be cast in precise mathematical forms. This is why the approach of computing safety stocks and order points based on service level considerarions /4/ch.6, /5/ seems to be more practical and, possibly, with more managerial appeal.

This paper proposes to analyze model 1 above, from the computational viewpoint, to find an efficient way to solve it using numerical methods, and then compare the results with the solution given by the authors.

2. HMMS MODEL FOR COMPUTING LOT SIZES UNDER AGGREGATE CONSTRAINT

2.1. The model

$$\min \sum_{i=1}^{n} \left(\frac{C_{Fi}S_i}{Q_i} + \frac{C_{Ii}Q_i}{2} \right)$$

s.t. (1)

$$\sum_{i=1}^{n} u_i \frac{Q_i}{2} = I_Q$$

where:

i = denotes item i, i=1,2,...,n;

C_{Fi} = setup cost for a lot ;

S_i = forecasted sales rate in units per period of time;

Q_i = lot size in units of product;

C_{Ii} = cost of holding one unit of inventory one period of time;

n_i = factor for converting units of the i'th product to the corresponding number of common units;

I_Q = aggregate inventory in the common unit.

Since the purpose of the paper is to study the solution to the model, it was left in the form presented by the authors. It is clear that if we also consider the problem of safety stocks the constraint in (1) should be expressed in terms of aggregate production P:

$$\sum_{i=1}^{n} u_i Q_i = P \tag{2}$$

rather than I_Q. This is because aggregate cycle stock cannot be computed unless we know the aggregate safety stock, which in turn

can be computed (under service level considerations) only after Q_i's are known. Fortunately the method of solving (1) remains unchanged if (2) becomes the constraint.

By using the Lagrangian multiplier method in (1) we transforme the constrained minimization into an unconstrained one.

$$\min L = \sum_{i=1}^{n} \left(\frac{C_{Fi}S_i}{Q_i} + \frac{C_{Ii}Q_i}{2} \right) + \lambda \left(I_Q - \sum_{i=1}^{n} \frac{u_iQ_i}{2} \right)$$

The first-order conditions for a minimum are $\frac{\partial L}{\partial Q_i} = 0$ yielding:

$$Q_i = \sqrt{\frac{2C_{Fi}S_i}{C_{Ii} - u_i}} \qquad (3)$$

Replacing Q_i into the constraint in (1) leads to an equation whose only unknown is λ:

$$\sum_{i=1}^{n} \frac{u_i}{2} \sqrt{\frac{2C_{Fi}S_i}{C_{Ii} - \lambda u_i}} = I_Q \qquad (4)$$

Since n is usually large, equation (4) is impossible to be solved exactly for λ.

2.2. <u>Authors' solution</u>

We are interested here in the general case of (3), when no special simplifying assumptions are made such as : constant sales composition, identical holding costs.

The first solution proposed by the authors is a <u>graphical method</u>. By drawing a smooth eye-fitted curve through a few plotted points, one can obtain the graph of the relation between I_Q and λ for any fixed set of salesrates $S_1, S_2, \ldots, S_n$. Given a certain aggregate inventory I_Q^* ,the value of λ can be found from the graph. This method is rather inaccurate and can lead to very large errors in the estimated λ for small values of I_Q^o as

the curve goes asymptotically to 0.

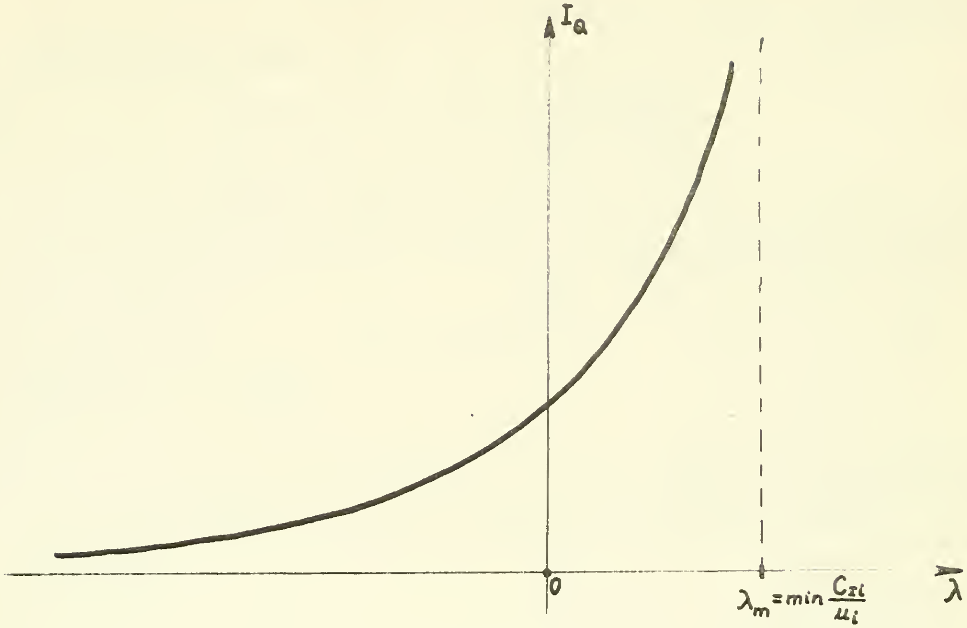


Fig.1

Another general solution is by <u>linear approximation</u> /3/p.193 .

$$\lambda = \lambda^o + A^o\left(I_Q^* - \tfrac{1}{2}I_Q^o - \tfrac{1}{4}\sum_{i=1}^{n} \frac{u_i Q_i^o}{S_i^o} S_i \right) \tag{5a}$$

where:

-we choose λ^o, S_i^o as an expansion point,

-compute Q_i^o, I_Q^o (for given λ^o, S_i^o) from equations (3), (4) respectively,

-I_Q^* is the constraining aggregate inventory, and S_i the forecasted demand,

-and $$A^o = 2\left[\sum_{i=1}^{n} \frac{u_i^2 Q_i^{o3}}{4C_{Fi} S_i^o}\right]^{-1}$$

In general, the approximation of equation (5a) could be improved by considering the <u>differential change in square root of the sales rates</u> rather than in sales rates. In this case

$$\lambda = \lambda^{0} + A^{0} \left(I_Q^{*} - \sum_{i=1}^{n} \frac{u_i Q_i^{0}}{2} \sqrt{\frac{S_i}{S_i^{0}}} \right) \qquad (5b)$$

A better estimate however is obtained if solution λ of (5a) or (5b) is further refined by a <u>logarithmic approximation</u>:

$$\lambda' = \lambda_m \left(1 - e^{-\frac{\lambda}{\lambda_m}}\right) \qquad (6)$$

where $\lambda_m = \min \frac{C_{Ii}}{u_i}$.

If, for instance $\lambda^{0} = 0$, then Fig.2 shows how the linearly approximated λ and the logarithmically approximated λ' are displayed:

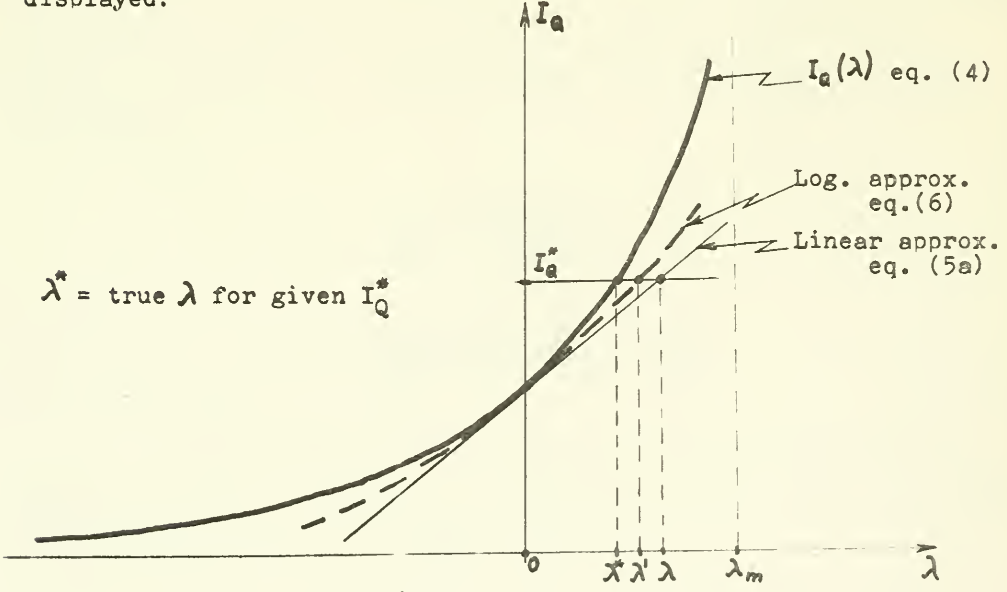


Fig.2

It is obvious that both solutions by linear and logarithmic approximation are sensitive to the choice of the starting

point (λ^o, S_i^o). The authors warn about this /3/p.194: "A convenient point of expansion is that characterized by the average sales rates and λ^o=0... . However, this is not always an appropriate point of expansion". No solution is offered as to how to choose λ^o when λ^o=0 is not appropriate. Thus, although the final formulas (5a), (5b), (6) are simple, the choice of λ^o may require iterative extensive analysis in order to obtain a fairly good estimate of λ^*.

It is apparent that a more accurate and efficient method is required for the solution of (4). The graph of $I_Q(\lambda)$ is shown in Fig.1. Given a certain aggregate inventory I_Q^* the problem is that of finding the root of an equation. However, it is easy to transform this root-finding problem into a maximization/minimization which has the advantage that some of the many optimum seeking methods available in the literature could be applyed to obtain the solution λ^*.

Let :

$$\psi(\lambda) = \sum_{i=1}^{n} \frac{u_i}{2}\sqrt{\frac{2C_{Fi}S_i}{C_{Ii}-\lambda u_i}} - I_Q^* \qquad (7)$$

$$f(\lambda) = \left|\sum_{i=1}^{n} \frac{u_i}{2}\sqrt{\frac{2C_{Fi}S_i}{C_{Ii}-\lambda u_i}} - I_Q^*\right| \qquad (8)$$

It is obvious that a solution to (4) is obtained when $\psi(\lambda)$=0, which is equivalent to minimizing f(λ). Fig.3 shows the relationship among $I_Q(\lambda)$, $\psi(\lambda)$, and f(λ).

Thusour problem is now one of unconstrained optimization (minimization).

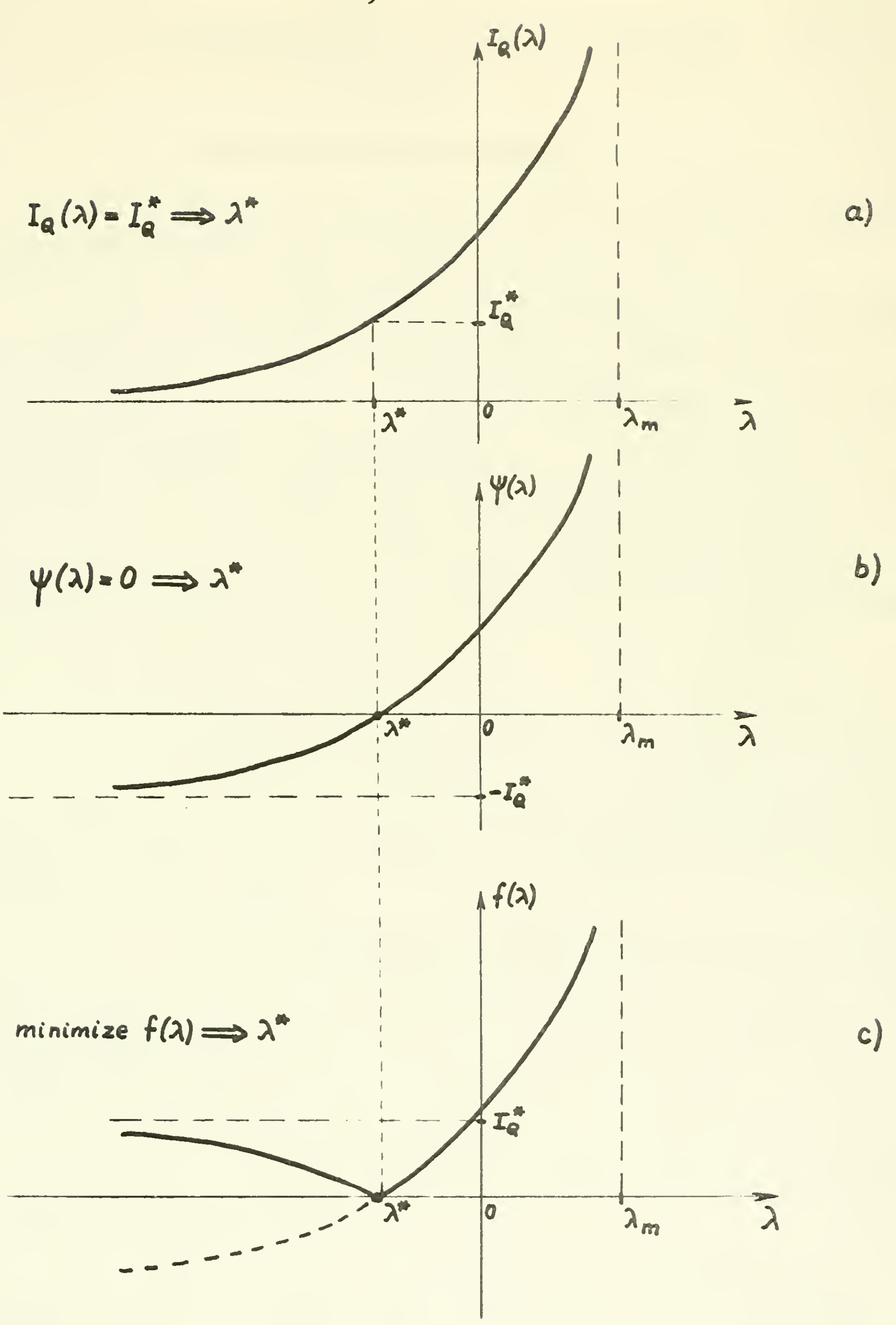

$I_Q(\lambda)$
$I_Q(\lambda) = I_Q^* \Longrightarrow \lambda^*$
a)
I_Q^*
λ^*
0
λ_m
λ
$\psi(\lambda)$
$\psi(\lambda) = 0 \Longrightarrow \lambda^*$
b)
λ^*
0
λ_m
λ
$-I_Q^*$
$f(\lambda)$
minimize $f(\lambda) \Longrightarrow \lambda^*$
c)
I_Q^*
λ^*
0
λ_m
λ

Fig.3

3. METHODS FOR UNCONSTRAINED OPTIMIZATION - A REVIEW -

3.1. Several variables problems

The literature presents a large set of optimum seeking methods for multivariate problems /10/, /11/, /6/, /7/.

Fletcher in /6/ groups these methods in 3 broad categories, according to their basic principle : gradient methods, direct search, and sums of squares. The problem to be considered is that of finding a local minimum (or if the function is unimodal the global minimum) of a function $f(\underline{x})$ where $\underline{x}$ is a vector of n variables $\underline{x}=(x_1,x_2,\ldots,x_n)$. Let $g(\underline{x})$ be the gradient of f with i-th element $\partial f/\partial x_i$, and G the matrix of second derivatives with i, j entry $\partial^2 f/\partial x_i \partial x_j$.

a) Gradient methods

- steepest descents in which the direction of search is $s=-g$. In practice the method improves $f(\underline{x})$ rapidly on the first few iterations and then gives rise to oscillatory progress and becomes unsatisfactory.

- Newton's method (or Taylor series method) assumes that the function may be approximated locally by its Taylor series up to the quadratic terms. Hence, the properties of quadratic functions are used directly to generate a direction of search $s=-G^{-1}g$. The convergence is rapid if $f(\underline{x})$ is adequately represented by the first two terms of its Taylor series. However, the method has various disadvantages :

- it requires the matrix of second derivatives to be provided and calculated at each iteration,

- it requires the solution of a set

of linear equationsto determine each direction of search .

Let's note that when G is locally singular the iteration breaks down; also if s turns out to be orthogonal to g no further progress is made.

- method of conjugate directions allows us to avoid calculating G; the cost we have to pay for this is that a larger number of iterations will be required. Since G^{-1} is no longer required the iteration cannot break down because of the singularity of G; also, the method ensures that the directions of search are downhill. If the vectors $s_1, s_2, \ldots, s_n$ have the property

$$s_i' G s_j = 0 \quad (i \neq j) \; ; \quad s_i' G s_j \neq 0 \quad (i = j)$$

with regard to a positive definite matrix G, then they are said to be conjugate. There are two ways in which a method can be made to generate conjugate directions: the parallel subspace method and the projection method.

All these methods find the minimum of a quadratic function in a given number of searches. For nonquadratic functions the methods can be applied iteratively. However, for nonquadratic functions a superior method has been developed by Davidon, called the variable metric method for minimization, in which a positive definite approximation H to G^{-1} is updated at each iteration, and is used to generate directions of search $s=-Hg$.

b) Direct Search Methods (/6/p.7, /7/p.26)

This set of methods is applied to functions whose derivatives are not available.

- alternating variable method - each variable is chosen in turn, all the others are kept constant and the extremum is obtained by one of the single variable search methods.

The method is very slow, highly oscillatory and usually fails to converge.

- Rosenbrock's modification (or pattern search as called in /8/p.B-347) is one of the most robust methods available for optimization when the derivatives are not available. This procedure has been obtained by imposing two modifications to the alternating variable method :

1) the first is to avoid the single variable optimization for each direction in turn. Instead a step of predetermined length is taken in each direction and these step lengths are modified after each calculation;

2) the second modification is to recognize that the alternating variable method takes a large number of very small steps and then to try to avoid this by realigning the axes. The axes are reoriented so that the first axis is along the most successful overall direction, the second axis along the next most successful direction and so on. The change of axes is performed by the well known Gram-Schmidt orthogonalization process.

- simplex method - the first step is to set up a regular "simplex" in an n-dimensional space, that is (n+1) points all equidistant from each other. The function is evaluated at those points and then the simplex set is altered systematically - dropping some points and adding others - until the region of the minimum is reached. Its precise location is found by interpolating a quadratic function at suitably chosen points. However, if the number of variables becomes large the method does not work so well.

It is interesting to point out the result obtained by Taubert /8/ in applying three search procedures, from the ones

enumerated above, for deriving decision rules for the aggregate scheduling problem. A 20 dimension response surface was searched for the minimum using: the method of conjugate direction, Davidon's variable metric method search, and the method of pattern search. /8/ reports that the pattern search exhibited the fastest convergence, while conjugate direction search was the slowest; Davidon's method yielded an average performance.

c) Sums of Squares - here the special case is considered in which $f(\underline{x})$ is the sum of squares of m nonlinear functions $g(\underline{x})$. The problem can be solved by minimizing $f(\underline{x})$ with one of the methods shown above. While this is usually the safest line, often much more rapid convergence can be obtained by taking into account the special nature of f.

When $m=n$ the problem of minimizing $f(\underline{x})$ is equivalent to that of solving a system of nonlinear equations $g(\underline{x})=0$. This is an interesting fact because, given a system of nonlinear equations $g_i(\underline{x})=0$, $i=1,2,..,m$, to solve it is equivalent to minimizing

$$f(\underline{x}) = \sum_{i=1}^{m} \left[g_i(\underline{x}) \right]^2$$

If no exact solution for minimizing $f(\underline{x})$ is available, the response surface $f(\underline{x})$ may be searched for its minimum and at least an approximate solution to the system of nonlinear equations can be found.

The above presentation gives a general, although not exhaustive by any means, view of the methods of searching for the minimum/maximum of functions of several variables. As a matter of fact, there are almost as many methods as there are researchers

in the field.

It is generally felt that dimensionality is probably the limiting factor in all these techniques. However, good progress has been made so far. For instance, /9/ch.7 reports a successful search conducted for finding the minimum of a 114-dimensional response surface. The cost was moderate (about 12 minutes on IBM 360/91) and the authors state: "we assume that we have not yet reached dimensionality limits so that the number of decision variables available is probably somewhat greater".

For the case of functions of one variable many of the above techniques can be applied. However, there is a special class of search methods developed for single variable problems, which are simpler than the previous one, and for particular cases (such as unimodal functions) probably more efficient.

3.2. Single variable problems

This section is meant to give an indication of the main methods used, with emphasis on the techniques that might be considered for solving the HMMS model.

3.2.1. Bracketing /7/

It is most important as a first step in a optimization to get a rough idea of where to look for an extremum; an useful idea is to find two values that bracket the extremum. Suppose the minimum of a function f(x) is sought and it is known that the minimum is located in the region $x \geqslant a$. Choose an increment ℓ and evaluate f at points $x_1=a$, $x_2=x_1+\ell$, $x_3=x_2+2\ell$, $x_4=x_3+4\ell$, $x_5=x_4+8\ell$, ..., that is doubling the increment at each stage.

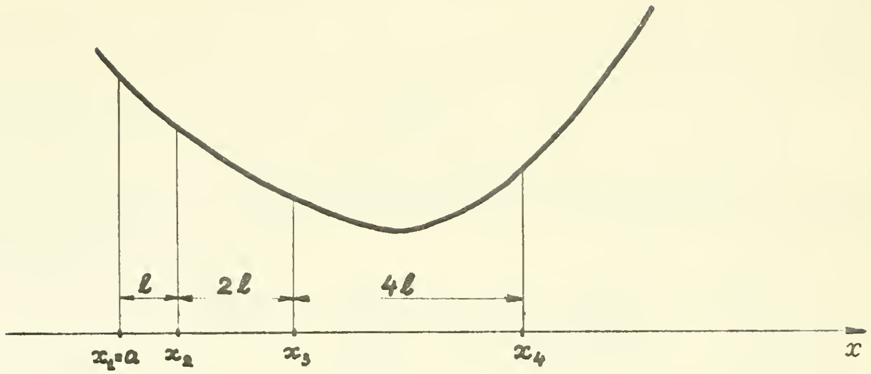


Fig.4

The evaluation is stopped if either the minimum is bracketed (Fig.4) or if $x_i > X$, where X is a suitable large constant chosen at the start of the calculation. The minimum is bracketed if at some stage $f(x_i) < f(x_{i-1})$ <u>and</u> $f(x_i) < f(x_{i+1})$; the bracket is (x_{i-1}, x_{i+1}). Of course, if we cannot decide the search direction at the start, both directions leaving x=a must be tried.

If the first derivative is available then if $f'(x_j) < 0$ and $f'(x_{j+1}) > 0$ the bracket for the minimum is (x_j, x_{j+1}).

If the value of f(x) decreases until X is reached it is usually assumed that the function is unbounded; if on the other hand the value of X is reached and the function is still increasing then f(a) is usually taken as the minimum value.

It was found that the method works well.

3.2.2. <u>Polynomial approximation</u> /7/

Once a bracket has been obtained for the extremum it is then required to obtain the extremum to any specified accuracy. One simple way of doing this is to use the information obtained by the bracketing procedure directly and approximate this information by a polynomial.

Suppose for instance that after a bracketing procedure developed as explained in 3.2.1. we stopped with $f(x_i) < f(x_{i+1})$, $f(x_{i+1})$. For simplicity let $z_1 = x_{i-1}$, $z_2 = x_i$, $z_3 = x_{i+1}$. A quadratic approximation for f can be written

$$f(x) = ax^2 + bx + c$$

Using the known information the result can be summarized in the matrix equation

$$\begin{bmatrix} f(z_1) \\ f(z_2) \\ f(z_3) \end{bmatrix} = \begin{bmatrix} z_1^2 & z_1 & 1 \\ z_2^2 & z_2 & 1 \\ z_3^2 & z_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

which can be solved for a, b, c. The minimum for the approximating quadratic is at $f'(x)=0$. Thus an estimate $f^* = f(x^*)$ for the extremum is now available. We can reiterate by identifying the best bracket from the values z_1, z_2, x^*, z_3, etc.

If the derivative is also available immediate advantage is obtained. Suppose in 3.2.1. we stopped with $f'(x_1) < 0$, $f'(x_2) > 0$, hence the bracketing interval (x_1, x_2). Since there are now four pieces of information: $f_1 = f(x_1)$, $f_1' = f'(x_1)$, $f_2 = f(x_2)$, $f_2' = f'(x_2)$ it is possible to approximate the function in the interval by a cubic:

$$f(x) = ax^3 + bx^2 + cx + d$$

$$f'(x) = 3ax^2 + 2bx + c$$

The coefficients are found by solving for a, b, c, d the following matrix equation :

$$\begin{bmatrix} f_1 \\ f_2 \\ f_1' \\ f_2' \end{bmatrix} = \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ 3x_1^2 & 2x_1 & 1 & 0 \\ 3x_2^2 & 2x_2 & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

The minimum of the approximating cubic is x^*, given by $f'(x)=0$. The gradients f_1', f_*', f_2' are compared and two new points are chosen to give gradients of opposite sign. The same procedure can then be repeated until sufficient accuracy is obtained.

This method works extremely well for most functions. However, some difficulty may arise if the function has a sharp peak.

3.2.3. Bolzano's root-finding method /11/ evaluates the function each time in the center of the remaining interval and eliminates half of the interval (whether the left half or the right half is eliminated depends on the evaluation outcome). After N evaluations of this sort, the ratio of initial L_o to final L_N interval is

$$\frac{L_o}{L_N} = 2^N \qquad (9)$$

Thus the number of observations needed to achieve a given reduction is evidently

$$N = 3.32 \log \frac{L_o}{L_N} \qquad (10)$$

This method is a contender for solving $\psi(\lambda)=0$ defined by equation (7) and shown in Fig.3.

Bolzano search /11/requires the evaluation of both the function and its first derivative. Each time a point is placed in the center of the remaining interval. Let for instance the bracketing interval be (x_1,x_2) and place point $x_3=\frac{x_1+x_2}{2}$. Evaluate $f_1=f(x_1)$, $f_2=f(x_2)$, $f_3=f(x_3)$, $f_1'=f(x_1)$, $f_2'=f(x_2)$, $f_3'=f(x_3)$ We had initially $f_1'<0$, $f_2'>0$ (since we are minimizing). The remaining interval is (x_1,x_3) if $f_3'>0$, and (x_3,x_2) if $f_3'<0$. Stop when the interval is small enough or when no further significant decrease in the objective function is achieved.

3.2.4. Direct search /10/, /7/.

This class of methods is concerned with optimizing when the derivative is not known or is complicated/inconvenient to be used. The idea is that once a bracket has been obtained the aim is to progressively reduce the length of the bracket until it is less than a prescribed limit, or until no significant improvement in the response function can be achieved.

Dichotomous search /10/ is similar to Bolzano search but it does not require the evaluation of the first derivative. Instead, two points rather than one are placed at a distance ε, symmetric to the center of the remaining interval.

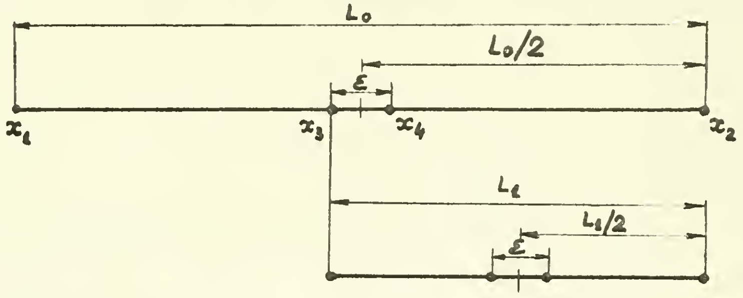


Fig.5

The evaluation of the function f(x) at each point is called an experiment. In Fig.5, after point x_3, x_4 have been determined we compute $f_3=f(x_3)$ and $f_4=f(x_4)$. Suppose $f_4 < f_3$; then we know that the minimum of f(x) lies somewhere in (x_3,x_2), hence L_1 is the remaining interval after the first set of experiments.

After N experiments (N must be even of course) we can locate the minimum within an interval of length:

$$L_N = L_o\left[\frac{1}{2^{N/2}} + \left(1 - \frac{1}{2^{N/2}}\right)\mathcal{E}\right] \qquad (11)$$

The interval $\mathcal{E}$ should be as small as possible; it is bounded from below by the requirement that two outcomes be distinguishable. It is important to point out that although the resolution is negligible compared to the original interval of uncertainity L_o, it is often a large fraction of the final interval L_N if the search is at all efficient.

<u>Golden section and Fibonacci search</u> /7/ are more powerful than the dichotomous search technique.

Suppose (a_1,a_2) brackets a required minimum of the function f(x). The points a_3, a_4 are symmetrically placed in this interval, so that

$$a_3 = \alpha_1 a_1 + (1 - \alpha_1)a_2$$
$$a_4 = (1 - \alpha_1)a_1 + \alpha_1 a_2 \qquad \frac{1}{2} < \alpha < 1 \qquad (12)$$

and this division is illustrated in Fig.6.

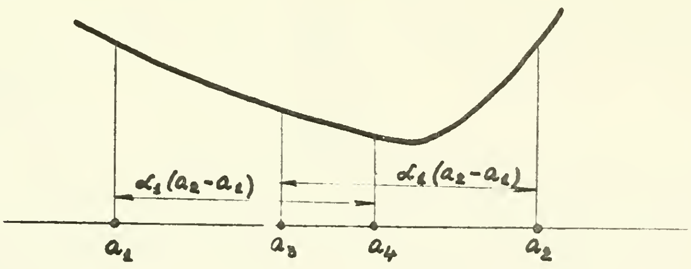


Fig.6

Suppose now that $f(a_4) < f(a_3)$; in this case (a_3,a_2) brackets the minimum.

Let's take now the remaining reduced interval (a_3,a_2) and divide it again. Since the number of functions evaluations

must be reduced to a minimum it would be very convenient to use the remaining point a_4 in a further symmetrical division of the reduced interval.

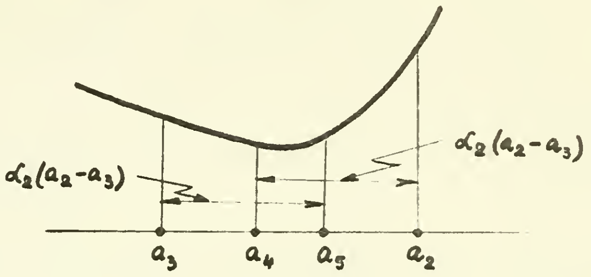


Fig.7

Indeed, place a new point a_5 symmetrical to a_4 in the interval (a_3,a_2) Fig.7.

$$\begin{aligned} a_5 &= (1-\alpha_2)a_3 + \alpha_2 a_2 \\ a_4 &= \alpha_2 a_3 + (1-\alpha_2)a_2 \end{aligned} \tag{13}$$

Since a_4 is the same in (12) and in (13) it follows that:

$$(1-\alpha_1)a_1 + \alpha_1 a_2 = \alpha_2 a_3 + (1-\alpha_2)a_2$$

which yields (14)

$$\alpha_2 = \frac{1-\alpha_1}{\alpha_1}$$

The method can be continued in precisely the same way successive symmetric divisions being performed until the length of the interval is less than the required tolerance. The sequence of fractions α_1, α_2,... satisfy the recurrence relation

$$\alpha_{N+1} = \frac{1-\alpha_N}{\alpha_N} \tag{15}$$

The basic choice is how to satisfy (15) in the most convenient manner.

The <u>golden section</u> sets $\alpha = \alpha_1 = \alpha_2 = \alpha_3 = \ldots$ and solving (15) gives

$$\alpha = 0.6180335 \qquad (16)$$

The <u>Fibonacci search technique</u> uses the Fibonacci numbers and works with a <u>prespecified</u> number N of interval divisions (i.e. experiments). Fibonacci numbers satisfy :

$$F_o = F_1 = 1, \quad F_N = F_{N-1} + F_{N-2}, \quad N \geqslant 2 \qquad (17)$$

Some of these numbers are given in the table of Fig.8 for later use.

N	0	1	2	3	4	5	6	7	8	9	10	11
F_i	1	1	2	3	5	8	13	21	34	55	89	144

Fig.8

Taking

$$\alpha_i = \frac{F_{N-i}}{F_{N-i+1}} \qquad (18)$$

it is easily seen that the recursive relation (15) is satisfied. The sequence of α_i's for this method is :

$$\begin{aligned} \alpha_1 &= \frac{F_{N-1}}{F_N} \\ \alpha_2 &= \frac{F_{N-2}}{F_{N-1}} \\ &\vdots \\ \alpha_{N-1} &= \frac{F_1}{F_2} \end{aligned} \qquad (19)$$

In the final division with $\alpha_{N-1} = \frac{F_1}{F_2} = \frac{1}{2}$ it is seen that the two end points and the <u>midpoint</u> have been selected. These three values have already been calculated and it is usual to evaluate the function f(x) at a point close to the midpoint to decide which half to choose as the final interval.

A slightly modified version of Fibonacci search, where the last experiment consists of placing two points at a very small distance ε from each other rather than only one in the middle of the interval, is developed in /10/p.24.

It can be shown that, if one starts with a bracketing interval of length L_o and the number N of experiments is sufficiently large, the ratio of the reduced intervalsobtained after (N-1) experiments with golden section and Fibonacci search respectively is :

$$\frac{\text{golden section length}}{\text{Fibonacci length}} = 1.1708$$

Thus for large N the Fibonacci search gives a 17% better result than the golden section. Both methods are superior to the dichotomous search because at every step the point remaining in the reduced interval is used in a further symmetrical division; this feature leads to a better use of the information available after each experiment.

Indeed, after N experiments the initial interval L_o is reduced down to:

$$L_N = \alpha_1 \alpha_2 \ldots \alpha_{N-1} L_o$$

For the golden section

$$L_N = \frac{L_o}{(1.618)^{N-1}}$$

and with Fibonacci search

$$L_N = \frac{L_o}{F_N} \quad .$$

If these results are compared with the result of the dichotomous search (equation 11) it's obvious that the golden section and Fibonacci search perform much more efficiently.

For an unimodal function it can be proven that both

golden section and Fibonacci search technique always work.

Lattice search by Fibonacci technique /10/

Wilde /10/ raises the point that it may sometimes be advantageous to convert an ordinary continuous search into a lattice search artificially.Indeed, the result of a continuous Fibonacci search willbe an interval that contains the optimum. When one is expected to make a decision based on the results of a search it is a bit frustrating to be confronted with an interval of uncertain ty. A precise point would be preferable, since a specific decision is called for. One could, of course, choose a point at random in the final interval of uncertainity, but most people would prefer a point where a measurement had already been made. Thus, to avoid these difficulties, the original problem can be converted into a lattice problem by placing a number of points in the bracketing interval so that the final answer will be a specific point on which a firm decision can be based.

Suppose it has been decided that a number N of experiments will be performed to search by Fibonacci a bracketing interval. Partition the interval into F_N units using F_N-1 points, not necessarilly equidistant. These points form a lattice. Let's associate the lattice points with the integers 1 through F_N-1. Thus, we are dealing here with an original interval of length F_N units. According to relation (19) and Fig.6 the first two experiments should be placed at a distance of $\alpha_1 F_N=F_{N-1}$ so that the first two experiments will coincide with points of the lattice, namely point number F_{N-1} and point number $F_N-F_{N-1}=F_{N-2}$.

Since the length of the interval remaining is also a

Fibonacci number, we see that the third experiment will also fall on one of the lattice points. This procedure may be continued until N-1 experiments have been used up and the length of the interval of uncertainity is down to $\alpha_1 \alpha_2 \ldots \alpha_{N-2} F_N = F_2 = 2$ units. The sequence is stopped and the unique point left inside the two unit long interval is compared to the end points and the best will be the estimate of the optimum, on which then all decisions will be based

Let us emphasize again that, <u>in order for this technique to work properly, the initial bracketing interval must be partitioned by a number of points equal to a Fibonacci number less one</u>.

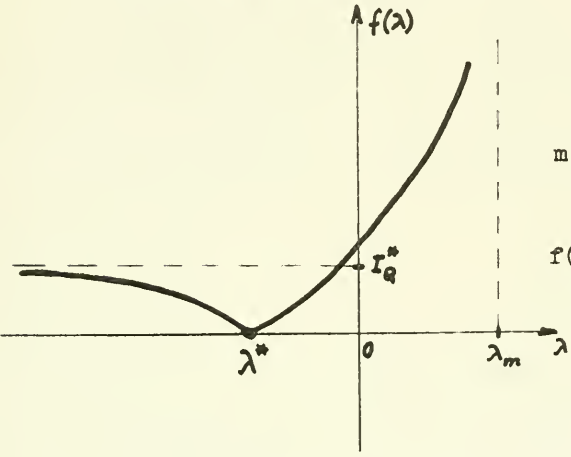
4. <u>SOLUTION TO HMMS MODEL BY LATTICE SEARCH WITH FIBONACCI TECHNIQUE</u>

From all the search methods reviewed in section 3 Fibonacci technique gurantees the largest interval reduction in a given number of steps. Moreover it does not require the use of derivatives, a fact that is advantageous in our case because the repeated evaluation of the derivative of (8) would require additional computing effort given the fact that it is computationally more complicated than the original function.

Thus, the task of this section is the search for the minimum of function (8) whose graph and equation will be repeated here for convenience (Fig.8).

We will use the same example as the one given by the authors in /3/p.196 in order to be able to compare performance. Assume that the estimates of the setup costs and the costs of holding inventory for each product are reviewed and revised

annually. Forecasts are also made of the average monthly sales rate, S_i, for each product for the coming year. The aggregate inventory levels are planned with view to both labor requirements and costs associated with inventory. The aggregate inventory will therefore be measured in labor hours by multiplying the units of each product by the conversion factor u_i, which has the dimension labor hours per item. The relevant data are summarized in the table of Fig. 9.



minimize $f(\lambda) \Longrightarrow \lambda^*$

$$f(\lambda) = \left| \sum_{i=1}^{n} \frac{u_i}{2} \sqrt{\frac{2C_{Fi}S_i}{C_{Ii}-\lambda u_i}} - I_Q^* \right| \qquad (8)$$

Fig.8

	Product			Units
	1	2	3	
C_{Ii}	1	1	2	\$/item-month
C_{Fi}	10	10	30	\$/lot
u_i	1	5	2	hours/item
S_i	2000	2000	10000	items/month

Fig.9

The point of expansion for linear/logarithmic

approximation was chosen $\lambda^0=0$ and the averages sales rates for each item. The table in Fig.10 shows alternative estimates of λ^* and the error in I_Q compared to the imposed aggregate level I_Q^* (I_Q is computed using the estimate of λ^*).

	$I_Q^*=$ 800	I_Q resulting from λ^*	Error %	$I_Q^*=$ 1500	I_Q resulting from λ^*	Error %
λ^* by linear approximation (5a)	-0.22	931	16.4	0.22	*	*
λ^* by logarithmic approximation (6)	-0.41	832	4	0.14	1611	7.4
λ^* by graphical method	-0.45	815	1.9	0.12	1481	-1.3

*infeasible since the estimate of .22 exceeds $\lambda_m = \min \frac{C_{Ii}}{u_i} = .20$

Fig.10

Both linear and logarithmic approximations overstate λ^* and may lead to considerable error, even to infeasibility. The graphical estimate is more accurate but it does not seem to be very appealing from a practical point of view because in order to get a fairly good curve one needs a large number of points to be evaluated; moreover, as shown in section 2, given the asymptotic characteristics of $f(\lambda)$ the errors in reading off the graph for small values of I_Q^* are likely to be quite large.

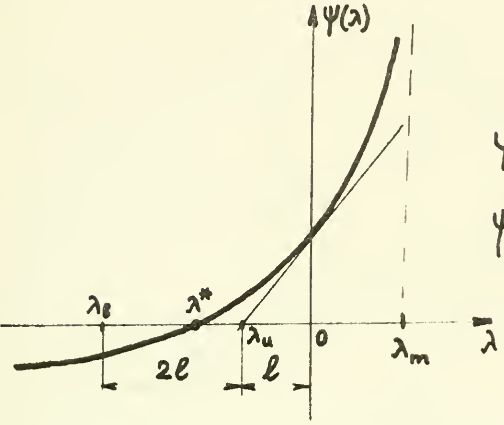
Let us now estimate λ^* using search techniques.

4.1. Bracketing λ^*

4.1.1. The case $I_Q^* < I_Q(0)$

In this case $\lambda^* < 0$. An upper bound on λ^* is easily obtained if we observe that the intersection between the tangent

to $\psi(\lambda)$ at $\lambda=0$ and the horizontal axis is always larger than λ^*. Let's call this upper bound λ_u. The reason behind the fact that we used $\psi(\lambda)$ rather than the searched response function $f(\lambda)$ to derive bounds will become clear in section 5. Anyhow, the brackets developed with $\psi(\lambda)$ are perfectly valid for $f(\lambda)$, given the equivalence shown in Fig.3.



$$\psi(\lambda)=0 \Longrightarrow \lambda^*$$

$$\psi(\lambda) = \sum_{i=1}^{n} \frac{u_i}{2}\sqrt{\frac{2C_{Fi}S_i}{C_{Ii}-u_i}} - I_Q^* \qquad (7)$$

Fig.11

A lower bound can be obtained using the bracketing technique described in 3.2.1. Use $\ell=-\lambda_u$ and go out from $-\lambda_u$ with a double increment 2ℓ (see Fig.11). Evaluate $\psi(\lambda_u-2\ell)$ and repeat the procedure (if necessary) with 4ℓ, 8ℓ etc. until $\psi(\lambda)$ becomes negative. Then stop with the lower bound λ_ℓ.

4.1.2. The case $I_Q^* > I_Q(0)$

In this case $0 < \lambda^* < \lambda_m$. The interval $(0, \lambda_m)$ is a possible choice for bracketing λ^*. However, for the porpose of computing an upper bound on the error of estimation (see section 6), this interval is unsuitable. An upper bound better than λ_m is required. Two subcases can be distinguished here :

a) λ corresponding to the intersection between the

tangent to $\psi(\lambda)$ at $\lambda=0$ and the horizontal axis is smaller than λ_m (Fig.12). In this situation the intersection is labeled λ_u, and a convenient bracketing interval is $(0, \lambda_u)$

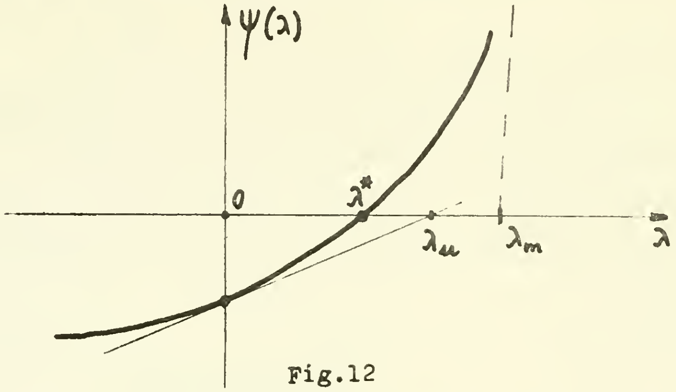


Fig.12

b) λ corresponding to the intersection between the tangent to $\psi(\lambda)$ at $\lambda=0$ and the horizontal axis is larger than λ_m (Fig.13).

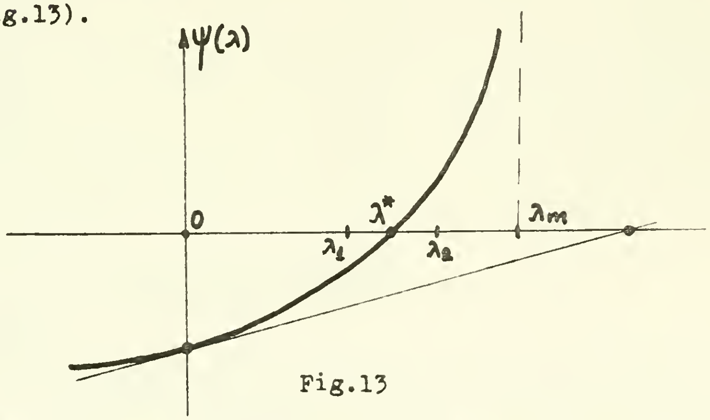


Fig.13

Then, a good bracketing procedure is a binary search /12/p.82.
The algorithm is the following :

Let k=1, $\lambda_0=0$

Step 1 - Evaluate $\psi(\lambda)$ at

$$\lambda_k = \lambda_{k-1} + \frac{\lambda_m - \lambda_{k-1}}{2}$$

If $\psi(\lambda_k) < 0$ go to step II.

If $\psi(\lambda_k)=0$ stop with $\lambda^* = \lambda_k$.

If $\psi(\lambda_k)>0$ stop with $\lambda_u = \lambda_k$ and $\lambda_\ell = \lambda_{k-1}$

Step 2 - Let k=k+1. Go to step I.

In Fig. 13 the algorithm terminated after two evaluations, yielding (λ_1, λ_2) as the bracketing interval.

Let's note that the binary search is actually the bracketing technique of section 3.2.1. applyed in reversed order; this was possible because we started out with a finite interval $(0, \lambda_m)$.

4.2. Estimating λ^*

Once the bracketing interval has been obtained, Fibonacci search technique will be applyed to an artifficially constructed lattice. Namely we decide first to perform N Fibonacci experiments; consequently F_{N-1} points will partition the initial interval into F_N units, and the technique is applyed to this lattice. The issue of how to choose N will be addressed in section 6.

4.3. Numerical examples

Example 1

Consider the example solved by the authors for an aggregate inventory of $I_Q^*=800$. All relevant data are given in the table of Fig. 9

In our particular case we have:

$$\psi(\lambda) = \frac{1}{2}\sqrt{\frac{40000}{1-\lambda}} + \frac{5}{2}\sqrt{\frac{40000}{1-5\lambda}} + \sqrt{\frac{600000}{2-2\lambda}} - 800$$

The tangent to $\psi(\lambda)$ at $\lambda=0$ is:

$$\psi - 348 = 1573.86\,\lambda$$

Applying the bracketing procedure shown in 4.1.1. one obtaines:

$$\lambda_u = -.22 \quad ; \quad \lambda_\ell = -.66$$

hence the inteval within which to search for λ^* is (-.66,-.22); the response function whose minimum we are searching for, is :

$$f(\lambda) = |\psi(\lambda)|$$

and it was pictured in Fig. 8.

Choose, for instance, to perform N=5 Fibonacci experiments. Consequently, the interval (-.66,-.22) will be partitioned into F_N=8 units using F_{N-1}=7 points equally spaced. However, there is no special requirement that the points be equidistant; consequently, if we have any suspicion that λ^* might lie in a certain subinterval of the bracketing interval we might want to distribute the points closer in that subinterval and further apart in the rest.

The increment of the lattice is h = $\frac{.44}{8}$ =.055.

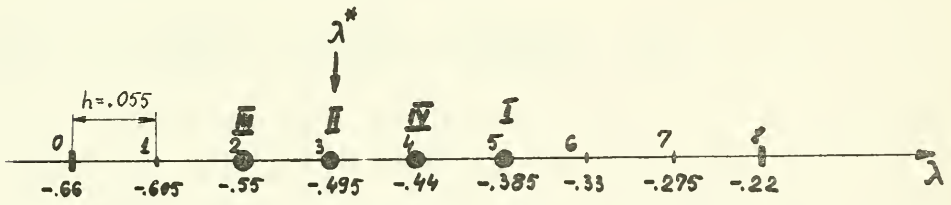


Fig.14

<u>First and second Fibonacci experiments</u>

Place two points at a distance $\alpha_1 F_N = F_{N-1}$ units from the ends of the interval; F_{N-1} = 5, so the two points are point no.5 (λ=-.385) and point no.3 (λ=-.495).

$$f(-.495) = 2.0324, \quad f(-.385) = 42.7343$$

Third experiment

The remaining interval is $(-.66,-.385)$; place in the interval a point symmetric to $(-.495)$. This point will be $\lambda=-.55$.

$$f(-.55) = 21.53805$$

Fourth experiment

The remaining interval is $(-.55,-.385)$ and the point chosen $\lambda=-.44$

$$f(-.44) = 19.27695$$

Fifth experiment

The remaining interval is $(-.55,-.44)$ and it contains only one point $\lambda=-.495$. This is the best point in the lattice, so it is chosen as our estimate of λ^*.

$$\underline{\lambda^* = -.495}$$

Fig. 14 shows the five Fibonacci experiments indicating the sequence in which they have been chosen.

Example 2

Consider the same data except for the aggregate inventory level which should be now $I_Q^*=1500$.

$$\psi(\lambda) = \frac{1}{2}\sqrt{\frac{40000}{1-\lambda}} + \frac{5}{2}\sqrt{\frac{40000}{1-5\lambda}} + \sqrt{\frac{600000}{2-2\lambda}} - 1500$$

The tangent to $\psi(\lambda)$ at $\lambda=0$ is:

$$\psi + 352 = 1573.86\,\lambda$$

The vertical asymptote for $\psi(\lambda)$ is $\lambda_m=0.2$. The intersection of the above tangent with the horizontal axis $\psi=0$ falls beyond $\lambda_m=0.2$ at $\lambda=.22$, hence this point is infeasible and can't be used as an upper bracket (see Fig.13). Apply then the binary search described in 4.1.2.b.

$\underline{k = 1}$, $\lambda_o = 0$

$$\lambda_1 = \lambda_o + \frac{\lambda_m - \lambda_o}{2} = .10$$

$$\psi(\lambda_1) = -110.1339 < 0$$

$\underline{k = 2}$

$$\lambda_2 = \lambda_1 + \frac{\lambda_m - \lambda_1}{2} = .15$$

$$\psi(\lambda_2) = 202.5537 > 0$$

Hence, the bracketing interval is $(\lambda_1, \lambda_2) = (\lambda_\ell, \lambda_u) = (.10, .15)$.

Let's choose to perform N=5 experiments. The corresponding partitioning of the interval is shown in Fig.15. After the search the best estimate turns out to be λ^*=.125. The Roman numerals in Fig.15 show the sequence in which the experiments have been placed.

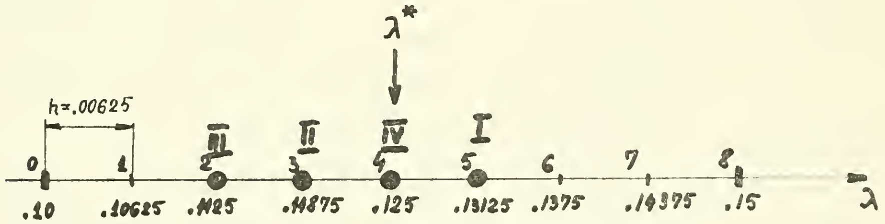


Fig. 15

In example 1 the estimated λ^*=-.495 yields an aggregate inventory I_Q=798, i.e. an error of 0.25% as compared with the imposed I_Q^*=800. In example 2 our estimate λ^*=.125 leads to I_Q=1509, i.e. an error of 0.6% relative to the constraining I_Q^*=1500.

The point which we would like to emphasize here is not

primarily the accuracy itself obtained above; this accuracy is more or less important if we look at it in connection with the accuracy of the input data (setup and holding costs, demand forecasts, conversion factors). The importantpoint is rather the small computational effort required to obtain this accuracy, and the fact that it can be easily improved by increasing the number of experiments (points) by just a few. Indeed, an initial interval of uncertainity can be reduced to less than one per cent of its original length after only eleven sequential experiments.

5. AN ALGORITHM FOR FINDING THE ROOT OF AN EQUATION BY LATTICE SEARCH WITH FIBONACCI NUMBERS

Fibonacci search technique is meant for finding the minimum/maximum of a function of a single variable (unimodality is desirable to ensure a successful search) This is why, although our original problem was to find the root of a complicated equation (7), we transformed it to obtain a peaked function $f(\lambda)$ (see Fig.3) on which the search has been performed.

In this section the original problem will be addressed. We propose a general purpose lattice - Fibonacci type algorithm for searching for the root of an equation within some previously determined bracketing interval. This method will be used to search the HMMS model and it will be clear that it requires a smaller number of function evaluations; the number of evaluations depends on where in the bracketing interval is the root located.

In Fig. 16 , (A,B) is a bracketing interval for λ^*; λ^* is at the same time the root of $\psi(\lambda)=0$ and the minimum of $f(\lambda)$. Suppose C,D are the first two lattice - Fibonacci experi-

ments placed in (A,B); then, according to the technique in section 4, the remaining interval will be (C,B) as $f(D) < f(C)$.

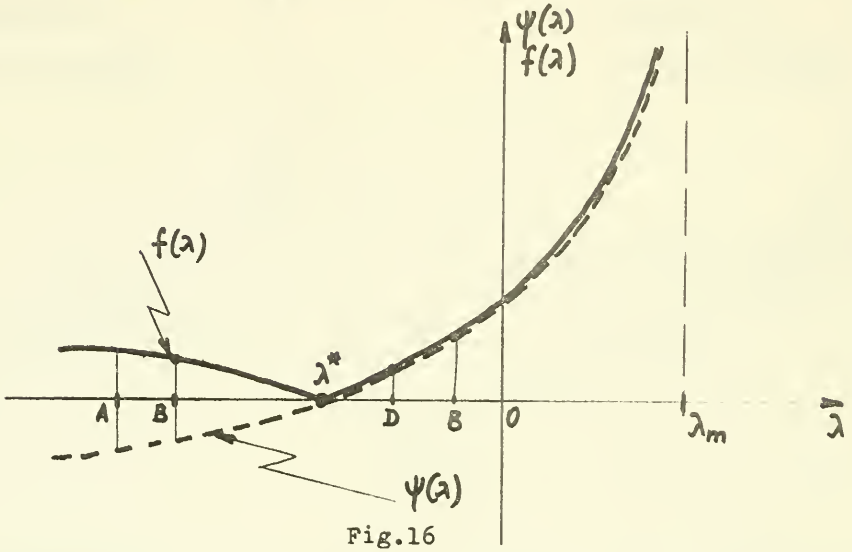


Fig.16

However, if we look at $\psi(\lambda)$ instead of $f(\lambda)$ the remaining interval would be (C,D) as $\psi(C)\psi(D) < 0$. Hence the idea of modifying the criterion of selecting the remaining interval, while keeping the rule of placing experiments according to the lattice - Fibonacci search method. We realize that the root λ^* could fall not only in the interval (C,D), but also in (A,C) or (D,B), as shown in Fig. 17, so after an experimental step the remaining interval could be either (A,C), (C,D) or (D,B). In order to further apply lattice - Fibonacci technique on the remaining interval it must contain a number of lattice points equal to a Fibonacci number less 1. Let's investigate (A,C), (C,D), (D,B) from this point of view.

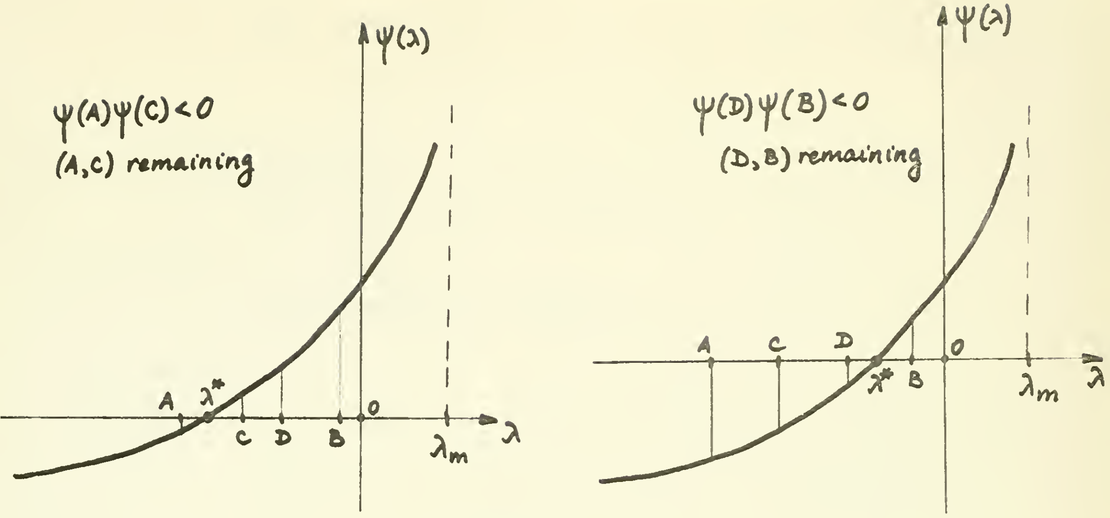


Fig.17

<u>PROPOSITION</u> Let the open interval (A,B) contain F_N-1 points, where F_N is a Fibonacci number. Suppose the first two experiments $C < D$ have been placed in (A,B) according to the lattice - Fibonacci search technique. Then, each open interval (A,C), (D,B) will contain F_p-1 points, and the open interval (C,D), F_q-1 points, where F_p, F_q are Fibonacci numbers.

<u>Proof</u> Points A, B, C, D, are displayed like in Fig 16. Let $N_{\gamma\delta}$ denote the number of points contained in some open interval (γ,δ).
According to 3.2.4. point D is the F_{N-1}-th point from A, and C is the F_{N-1}-th point from B. Thus

$N_{AD} = F_{N-1} - 1$, $N_{CB} = F_{N-1} - 1$.

Consequently :

$$N_{AC} = N_{AB} - N_{CB} - 1 = (F_n - 1) - (F_{N-1} - 1) - 1 =$$
$$= F_N - F_{N-1} - 1 = F_{N-2} - 1$$

Similarly :

$$N_{CD} = N_{AD} - N_{AC} - 1 = (F_{N-1} - 1) - (F_{N-2} - 1) - 1 =$$
$$= F_{N-3} - 1 ,$$

and clearly $N_{DB} = N_{AC}$. ||

<u>COROLLARY</u> - Let $\psi(\lambda)$ be a continuous function of λ with a root contained in some bracketing open interval (A,B). Assume (A,B) contains F_n-1 points, where F_n is a Fibonacci number, and that the first two experiments $C < D$ have been placed in (A,B) according to the lattice - Fibonacci search technique.

If the next (reduced) bracketing interval is chosen such that $\psi(\gamma)\,\psi(\delta) < 0$, $\gamma, \delta \in \{A,B,C,D\}$ then $N_{\gamma\delta}$ is equal to a Fibonacci number less 1.

<u>Proof</u> - The proof follows directly from the above proposition coupled with figures 16, 17. ||

Of course, from all possible intervals (γ, δ) constructed such that $\psi(\gamma)\,\psi(\delta) < 0$, $\gamma, \delta \in \{A,B,C,D\}$ we will choose the one with the minimum $N_{\gamma\delta}$ for the purpose of efficiency.

<u>Summary</u> : In solving the HMMS model we will search for the unique root of $\psi(\lambda)$; a lattice will be constructed inside the initial bracketing interval using a number of points equal to a Fibonacci

number less one. At each step two experiments are placed according the method described in 3.2.4., and the remaining interval will be chosen as specified by the corollary above. The procedure will be then repeated with the remaining bracketing interval.

Example

For the purpose of comparison the example already solved in 4.3. will be tackled by the modified algorithm. Consider the data in the table of Fig.9, and an aggregate inventory of $I_Q^*=1500$.

The bracketing interval is (.10,.15)(see 4.3); we have already chosen N=5, so there will be $F_N-1=7$ points in (.10,.15).

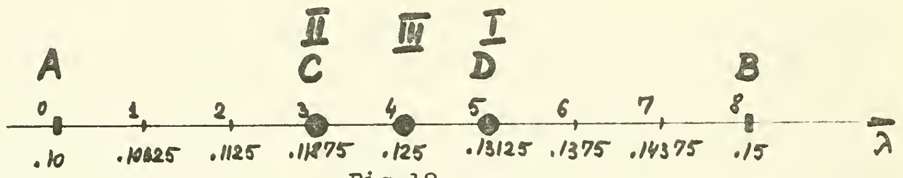


Fig.18

First and second Fibonacci experiments are placed at F_{N-1} units from both ends. So, C=.11875, D=.13125

$$\psi(.11875) = -25.5509$$
$$\psi(.13125) = 47.7335$$

Evidently, since $\psi(C)\,\psi(D) < 0$ the remaining interval is (C,D).

Third Fibonacci experiment - interval (C,D) only contains one point, so

$$\psi(.125) = 8.94 < |\psi(C)|, \psi(D) .$$

Hence $\lambda^*=.125$, and with the modified method 3 function evaluations were needed instead of 4 in section 4.3. The modified technique is faster in reducing the interval of uncertainity especially at the very beginning. This is easily seen if we compare

the remaining intervals obtained with the two methods applied to the lattice shown in Fig. 18:

Lattice-Fibonacci

Experiment performed	Remaining interval
λ=.13125 λ=.11875	(.10,.13125)
λ=.1125	(.1125,.13125)
λ=.125	(.11875,.13125)
-	Solution

Modified technique

Experiment performed	Remaining interval
λ=.13125 λ=.11875	(.11875,.13125)
λ=.125	Solution

The speed of convergence of the proposed technique is influenced by the position of the root in the bracketing interval: the fastest reduction in the uncertainity interval is obtained when the root lies in the central part of the bracketing interval. The results of two examples worked out on a lattice divided into 89 intervals (units) are shown below. In the first example it was assumed that the root is located in the central part, while in the second the root was located at the left extreme.

Example with root centrally located

Lattice-Fibonacci

Experiment performed	Remaining interval, in units
I, II	55
III	34
IV	21
V	13
VI	8
VII	5
VIII	3
IX	2
X	Solution

Modified technique

Experiment performed	Remaining interval, in units
I, II	21
III, IV	5
V, VI	2
VII	Solution

Example with root at the extreme left end

Lattice-Fibonacci		Modified technique	
Experiment performed	Remaining interval, in units	Experiment performed	Remaining interval, in units
I, II	55	I, II	34
III	34		
IV	21	III, IV	13
V	13		
VI	8	V, VI	5
VII	5		
VIII	3	VII, VIII	2
IX	2	IX	Solution
X	Solution		

The advantage of the fast convergence at the beginning of the search could be exploited by checking the uncertainity interval after each iteration. We do not have to perform all the experiments required by a Fibonacci search; we can stop whenever the uncertainity interval is accurate enough.

Example - Consider again Fig.18, where the initial bracketing interval is (.10,.15). We place 2 points in the interval: λ_1=.11875, λ_2=.13125 and the remaining interval in the modified technique is (.119,.13). Suppose this is sufficiently accurate, so stop with λ^* (.119,.13).

In order to obtain the same interval of uncertainity the original Fibonacci technique would have required 4 function evaluations at : λ_1=.119, λ_2=.13, λ_3=.138, λ_4=.126.

It is also apparent from the results presented above

that a smaller number of reduced intervals must be kept track of, which reduces computer time.

Thus, we will resort to the modified lattice-Fibonacci technique in the computer program for solving the HMMS model, using as a response function $\psi(\lambda)=0$.

6. CHOOSING THE NUMBER OF POINTS IN THE BRACKETING INTERVAL

There are two requirements to be met, and which place bounds on the number of points dividing the initial bracketing interval:

a)- the requirement that two adjacent function evaluations be distinguishable; this places an upper bound on the number of points in the interval.

b)- the accuracy of the result which is better if the number of points is larger; this requirement sets the lower bound on the number of points .

Let (A,B) be the initial bracketing interval and N_{AB} the number of points partitioning the open interval (A,B). Then /10/p.37:

$$N_{AB} \leq F_N - 1$$

where

N = the integer part of $4.785 \log \frac{1}{\varepsilon} - 0.328$

ε = the minimum spacing for which two outcomes are distinguishable

F_N = Fibonacci number

However, given the available modern computing capability it is probably unlikely for this upper bound to be constraining in the

case of a production problem like the HMMS model. In such cases we expect the cost of the computational effort to place an upper bound on the number of points. Thus, we willtend to select the smallest number of points allowed by the accuracy requirement.

One way to approach this problem is the following: given the lattice increment h and the fact that our estimate of λ^* is one of the lattice points, we know that we can not be off the true root by more than h. The change in the aggregate inventory corresponding to a variation of h in λ can be approximated by $\frac{dI_Q}{d\lambda}h$, for h small. As we are interested in the proportional change in aggregate inventory, we will use :

$$E_I = \frac{1}{I_Q}\frac{dI_Q}{d\lambda}h \tag{22}$$

where E_I is the proportional error in aggregate inventory.

Clearly, setting an upper bound on the error E_I is equivalent to limitting h from above and N_{AB} from below. But E_I is a function of λ and obviously $E_I(\lambda^*)$ can not be computed before λ^* is found. How should E_I be computed in order to determine the minimum number of points needed ?

$$E_I = \frac{1}{I_Q}\frac{dI_Q}{d\lambda}h = \frac{\sum_{i=1}^{n}\frac{u_i^2}{2}\frac{\sqrt{2C_{Fi}S_i}}{(C_{Ii}-\lambda u_i)^{3/2}}}{\sum_{i=1}^{n}u_i\frac{\sqrt{2C_{Fi}S_i}}{(C_{Ii}-\lambda u_i)^{1/2}}}\cdot h \tag{23}$$

Analyzing expression (23) we find that $E_I(\lambda)$ increases monotonically from 0 to ∞, for $-\infty < \lambda \leq \min \frac{C_{Ii}}{u_i}$. Consequently, given an interval $(\lambda_\ell, \lambda_u)$ bracketing λ^*, where λ_ℓ, λ_u are

the lower, and upper ends of the interval, respectively,

$$E_I(\lambda_u) > E_I(\lambda^*) \quad . \tag{24}$$

Hence, the error computed at λ_u is an upper bound on the error in the aggregate inventory corresponding to the estimated λ^*.

If L_o is the initial bracketing interval then $h = \frac{L_o}{F_N}$ so taking (22) and (24) into account we can find the lower bound on F_N:

$$F_N \geq \frac{1}{I_Q} \frac{dI_Q}{d\lambda} \frac{L_o}{E_I(\lambda_u)} \tag{25}$$

where all elements of the right hand side are computed at $\lambda = \lambda_u$, and $E_I(\lambda_u)$ is specified by the user of the model.

Example

Consider the data shown in the table of Fig.9 and an aggregate inventory level of $I_Q^* = 1500$. In section 4.3 the bracketing interval for λ^* was found to be (.10,.15). In how many units should we partition this interval ?

Assume, for instance, that we want to limit the error in aggregate inventory to 5%, i.e. 0.05. Consequently we set

$$E_I(\lambda_u) = E_I(.15) = 0.05 \quad .$$

$$I_Q(.15) = 1702.5$$

$$\frac{dI_Q}{d\lambda}(\quad =.15) = 10356.3$$

$$L_o = .05$$

Applying (25) we obtain

$$F_N \geq 6.083$$

Since F_N must be a Fibonacci number we look it up

in the table of Fig.8 and find $F_N = 8$ corresponding to 5 Fibonacci experiments. The number of points in the bracketing interval will be $F_N - 1 = 7$.

7. SOLUTION BY BOLZANO'S ROOT FINDING SEARCH METHOD

As shown in 3.2.3., after N evaluations of the response function, the initial bracketing interval L_o can be reduced to $L_o/2^N$ with Bolzano's method. If we have dealt with a continuous λ, after comparing 2^N with Fibonacci numbers in Fig.8 it would have been quite obvious that Bolzano's method achieves a greater interval reduction in a given number of iterations N. However, for reasons shown on page 23, we preferred to turn our problem into a discrete version, in which case it is not immediately clear which method performs better.

The general approach with Bolzano's technique is similar to what we did so far:

- set the maximum acceptable relative error in aggregate inventory,

- determine the number of intervals in which the initial bracketing interval must be partitioned; the number of intervals should be equal to a power of two, so the problem is that of finding the smallest k in:

$$2^k \geq \frac{1}{I_Q} \frac{dI_Q}{d\lambda} \frac{L_o}{E_I(\lambda_u)} \quad , \qquad (26)$$

- apply Bolzano search on the lattice just constructed.

A computer program has been set up to experiment with the two methods: the modified algorithm, and Bolzano's search. The following computer report shows the result of an

example run with the data in the table of Fig.9, for an imposed aggregate inventory of 800.

```
WHAT IS THE NUMBER OF ITEMS?
!3
WHAT IS THE AGGREGATE INVENTORY?
!800
WHAT IS HIGHEST ACCEPTABLE ERROR - PERCENTAGE ?
!3


MODIFIED LATTICE-FIBONACCI SEARCH ALGORITHM
===========================================

NUMBER OF ITEMS =  3
IMPOSED AGGREGATE INVENTORY =  800
MAXIMUM ADMISSIBLE ERROR =  3 %

LENGTH OF INITIAL BRACKETING INTERVAL =  .441869
ACTUAL NUMBER OF ITERATIONS =  4
ITERATIONS REQUIRED BY PURE FIBONACCI SEARCH =  6

ESTIMATED LAGRANGE MULTIPLIER =  -.492854
RESULTING AGGREGATE INVENTORY =  798.762
ACTUAL ERROR IN AGGREGATE INVENTORY =  -.15477 %

PRODUCTION PLAN
ITEM  1         163 UNITS
ITEM  2         107 UNITS
ITEM  3         448 UNITS

LATTICE SEARCH BY BOLZANO METHOD
================================

NUMBER OF ITERATIONS =  4

ESTIMATED LAGRANGE MULTIPLIER =  -.497103
RESULTING AGGREGATE INVENTORY =  797.189
ACTUAL ERROR IN AGGREGATE INVENTORY =  -.351364 %

PRODUCTION PLAN
ITEM  1         163 UNITS
ITEM  2         107 UNITS
ITEM  3         447 UNITS
```

It's clear that both the modified technique and Bolzano method had the same performance, while outperforming the pure Fibonacci search.

The explanation for the equal performance is the following: the lower limit on the number of partitioning intervals was a number less than 13. Consequently, by formula (25) a number of 13 divisions has been considered for the modified algorithm, while by formula (26) a number $2^4=16$ divisions were constructed for Bolzano's search. Thus, although Bolzano technique is in general the fastest, the discrete nature of our problem lead to some loss of efficiency. However, for larger intervals, or for a lower desired error, Bolzano's method performs better than the modified lattice-Fibonacci algorithm (see attached computer reports).

Conclusion - For inventory levels I_Q^* smaller but relatively close to I_Q ($\lambda=0$), and for I_Q^* larger than I_Q ($\lambda=0$) the initial bracketing interval is expected to be small, and there is no significant performance difference between the modified lattice-Fibonacci search and Bolzano's technique. Both perform extremely efficiently. For I_Q^* considerably smaller than I_Q ($\lambda=0$) Bolzano's method performs better and is to be preferred.

8. A COMPUTER PROGRAM FOR THE MODIFIED LATTICE-FIBONACCI SEARCH TECHNIQUE AND FOR BOLZANO'S METHOD

A computer program has been written in BASIC to perform the search for the Lagrange multiplier in the HMMS disaggregation model. The program has three major parts:

- the code for finding the bracketing interval,
- the code for performing the search with the modified lattice-Fibonacci technique,

- the code for Bolzano's method,

and 5 subroutines for:

- the evaluation of aggregate inventory function for a given multiplier value,
- computing the derivative of the response function,
- finding the asymptote of the response function,
- computing a Fibonacci number for the initial partitioning of the bracketing interval,
- computing Fibonacci numbers for a given number of experiments.

The user supplies all relevant cost data, the number of items, the constraining aggregate inventory and the upper bound on the error in inventory. The general logic of the program follows the development provided in the paper, and is illustrated in the attached flowchart.

The program was meant to serve the purpose of the comparative study; consequently, the reports provide the necessary information to assess efficiency.

Some examples have been worked out and the reports are shown below. A general flowchart as well as the entire program listing are attached.

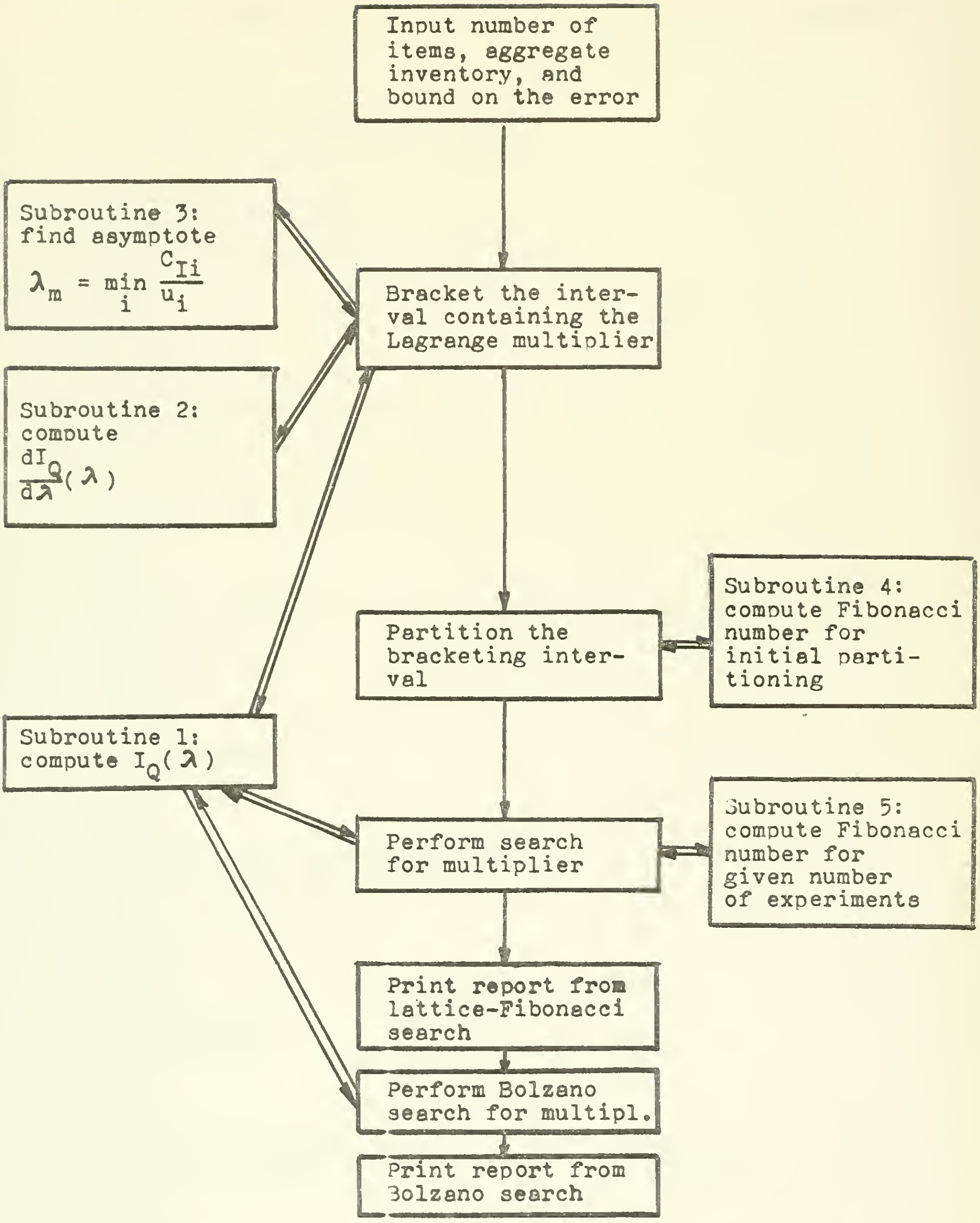
```
WHAT IS THE NUMBER OF ITEMS?
!3
WHAT IS THE AGGREGATE INVENTORY?
!800
WHAT IS HIGHEST ACCEPTABLE ERROR - PERCENTAGE ?
!2


MODIFIED LATTICE-FIBONACCI SEARCH ALGORITHM
===========================================

NUMBER OF ITEMS =  3
IMPOSED AGGREGATE INVENTORY =  800
MAXIMUM ADMISSIBLE ERROR =  2 %

LENGTH OF INITIAL BRACKETING INTERVAL =  .441869
ACTUAL NUMBER OF ITERATIONS =  5
ITERATIONS REQUIRED BY PURE FIBONACCI SEARCH =  7

ESTIMATED LAGRANGE MULTIPLIER =  -.494473
RESULTING AGGREGATE INVENTORY =  798.161
ACTUAL ERROR IN AGGREGATE INVENTORY =  -.229828 %

PRODUCTION PLAN
ITEM  1        163 UNITS
ITEM  2        107 UNITS
ITEM  3        448 UNITS

LATTICE SEARCH BY BOLZANO METHOD
================================

NUMBER OF ITERATIONS =  4

ESTIMATED LAGRANGE MULTIPLIER =  -.497103
RESULTING AGGREGATE INVENTORY =  797.189
ACTUAL ERROR IN AGGREGATE INVENTORY =  -.351364 %

PRODUCTION PLAN
ITEM  1        163 UNITS
ITEM  2        107 UNITS
ITEM  3        447 UNITS
```

```
WHAT IS THE NUMBER OF ITEMS?
!3
WHAT IS THE AGGREGATE INVENTORY?
!200
WHAT IS HIGHEST ACCEPTABLE ERROR - PERCENTAGE ?
!3


MODIFIED LATTICE-FIBONACCI SEARCH ALGORITHM
===========================================

NUMBER OF ITEMS =  3
IMPOSED AGGREGATE INVENTORY =  200
MAXIMUM ADMISSIBLE ERROR =  3 %

LENGTH OF INITIAL BRACKETING INTERVAL =  20.4736
ACTUAL NUMBER OF ITERATIONS =  12
ITERATIONS REQUIRED BY PURE FIBONACCI SEARCH =  13

ESTIMATED LAGRANGE MULTIPLIER =  -18.1975
RESULTING AGGREGATE INVENTORY =  199.963
ACTUAL ERROR IN AGGREGATE INVENTORY =  -1.83411E-02 %

PRODUCTION PLAN
ITEM  1        45 UNITS
ITEM  2        20 UNITS
ITEM  3        125 UNITS

LATTICE SEARCH BY BOLZANO METHOD
================================

NUMBER OF ITERATIONS =  9

ESTIMATED LAGRANGE MULTIPLIER =  -18.1966
RESULTING AGGREGATE INVENTORY =  199.968
ACTUAL ERROR IN AGGREGATE INVENTORY =  -1.61285E-02 %

PRODUCTION PLAN
ITEM  1        45 UNITS
ITEM  2        20 UNITS
ITEM  3        125 UNITS
```

REFERENCES

1. Hax, A.C., Aggregate Production Planning, A Survey, Sloan School of Management, M.I.T., OR 027-73, April 1974
2. Hax, A.C., and Meal,H.C., Hierarchical Integration of Production Planning and Scheduling, Management Science, Special Logistics Issue, January, 1975
3. Holt, C.C., Modigliani,F.,Muth,J.F., Simon,H.A., Planning Production, Inventories, and Work Force, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1960
4. Magee,J.F., Boodman,D.M., Production Planning and Inventory Control, McGraw-Hill, 1967
5. Hax, A.C., Technical Note on Inventory Control, Harvard Business School, 1971
6. Fletcher,R., Optimization, Academic Press, 1969
7. Burley, D.M.,Studies in Optimization, John Wiley & Sons, 1974
8. Taubert,W.H., A Search Decision Rule for the Aggregate Scheduling Problem, Management Science, Vol.14, No.6, February, 1968
9. Buffa, E.S., and Taubert,W.H., Production-Inventory Systems Planning and Control, Richard D. Irwin,1972
10. Wilde,D.J., Optimum Seeking Methods, Prentice-Hall, Inc., Englewood Cliffs, N.J.,1964
11. Wilde,D.J.,and Beightler,C.S., Foundations of Optimization, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1967
12. Donovan, J.J., Systems Programming, McGraw-Hill, 1972

APPENDIX I


Input number of items, aggregate inventory, and bound on the error
Subroutine 3: find asymptote $\lambda_m = \min_i \frac{C_{Ii}}{u_i}$
Bracket the interval containing the Lagrange multiplier
Subroutine 2: compute $\frac{dI_Q}{d\lambda}(\lambda)$
Partition the bracketing interval
Subroutine 4: compute Fibonacci number for initial partitioning
Subroutine 1: compute $I_Q(\lambda)$
Perform search for multiplier
Subroutine 5: compute Fibonacci number for given number of experiments
Print report from lattice-Fibonacci search
Perform Bolzano search for multipl.
Print report from Bolzano search

Flow chart showing the general features of the comparative search program

APPENDIX II : Computer program for the modified lattice-Fibonacci search and for Bolzano's root finding technique

```
 10  PRINT 'WHAT IS THE NUMBER OF ITEMS?'
 20  INPUT I
 30  FOR K=1 TO I
 40  READ C(K),H(K),U(K),S(K)
 50  NEXT K
 60  PRINT 'WHAT IS THE AGGREGATE INVENTORY?'
 70  INPUT A
 80  LET W2=1
 90  LET X=0
100  GOSUB 1430
110  IF A0=A THEN 1230
120  GOSUB 1490
130  REM * FIND INTERSECTION OF TANGENT AND HORIZONTAL AXIS *
140  LET X=(A-A0)/D
150  REM * FIND BRACKETING INTERVAL *
160  IF X<0 THEN 340
170  GOSUB 1550
180  IF X>=M THEN 230
190  LET E1=0
200  LET E2=X
210  GOTO 600
220  REM * BINARY SEARCH FOR FINDING BRACKETING INTERVAL *
230  LET E1=0
240  LET E2=E1+(M-E1)/2
250  LET X=E2
260  GOSUB 1430
270  LET Y=A0-A
280  IF Y=0 THEN 1250
290  IF Y<0 THEN 310
300  GOTO 450
310  LET E1=E2
320  GOTO 240
330  REM * BINARY SEARCH ENDS HERE *
340  LET E2=X
350  LET K=1
360  LET L=2^K*E2
370  LET X=X+L
380  GOSUB 1430
390  LET Y=A0-A
400  IF Y=0 THEN 1270
410  IF Y<0 THEN 440
420  LET K=K+1
430  GOTO 360
440  LET E1=X
450  REM ** PARTITION BRACKETING INTERVAL (E1, E2) **
460  PRINT 'WHAT IS HIGHEST ACCEPTABLE ERROR - PERCENTAGE ?'
470  INPUT B
480  LET X=E2
490  GOSUB 1430
500  GOSUB 1490
510  LET T=D*(E2-E1)/(A0*B/100
520  LET E3=E1
530  LET E4=E2
```

```
540  GOSUB 2260
550  LET L=E2-E1
560  GOSUB 1660
570  LET N0=N
580  LET W=0
590  REM *** MODIFIED LATTICE-FIBONACCI SEARCH ALGORITHM ***
600  LET X1=E1
610  LET X=X1
620  GOSUB 1430
630  LET Y1=A0-A
640  LET X2=E2
650  LET X=X2
660  GOSUB 1430
670  LET Y2=A0-A
680  IF F=2 THEN 710
690  IF F>2 THEN 830
700  GOTO 1130
710  LET X3=(X1+X2)/2
720  LET X=X3
730  GOSUB 1430
740  LET W=W+1
750  Y3=A0-A
760  IF Y3=0 THEN 1290
770  LET F=1
780  IF Y3<0 THEN 810
790  LET E2=X3
800  GOTO 600
810  LET E1=X3
820  GOTO 600
830  LET J=N-1
840  GOSUB 1780
850  LET X3=X1+G*(E2-E1)/F
860  LET X=X3
870  GOSUB 1430
880  LET W=W+1
890  LET Y3=A0-A
900  IF Y3=0 THEN 1290
910  LET X4=X2-G*(E2-E1)/F
920  LET X=X4
930  GOSUB 1430
940  LET W=W+1
950  LET Y4=A0-A
960  IF Y4=0 THEN 1310
970  IF Y4<0 THEN 990
980  GOTO 1010
990  IF Y3<0 THEN 1070
1000 GOTO 1090
1010 LET E2=X4
1020 LET N=N-2
1030 LET J=N
1040 GOSUB 1780
1050 LET F=G
1060 GOTO 600
```

```
1070  LET E1=X3
1080  GOTO 1020
1090  LET E1=X4
1100  LET E2=X3
1110  LET N=N-3
1120  GOTO 1030
1130  REM **THE INTERVAL IS 1 UNIT LONG AND CONTAINS NO POINT**
1140  REM * INTERVAL END WITH THE SMALLEST ERROR IN *
1150  REM * AGGREGATE INVENTORY IS MULTIPLIER ESTIMATE *
1160  LET Y1=ABS(Y1)
1170  IF Y1<Y2 THEN 1210
1180  REM * SEARCH IS COMPLETE; X0 IS MULTIPLIER ESTIMATE *
1190  LET X0=X2
1200  GOTO 1320
1210  LET X0=X1
1220  GOTO 1320
1230  LET X0=0
1240  GOTO 1320
1250  LET X0=E2
1260  GOTO 1320
1270  LET X0=X
1280  GOTO 1320
1290  LET X0=X3
1300  GOTO 1320
1310  LET X0=X4
1320  LET X=X0
1330  GOSUB 1430
1340  LET P=((A0-A)/A)*100
1350  FOR K=1 TO I
1360  LET Q(K)=(2*D(K)*S(K)/(H(K)-X*U(K)))^.5
1370  LET Q(K)=INT(Q(K))
1380  NEXT K
1390  IF W2=2 THEN 2050
1400  GOTO 1920
1410  REM ***** SUBROUTINES *****
1420  REM ** SUBROUTINE FOR COMPUTING AGGREGATE INVENTORY **
1430  LET A0=0
1440  FOR K=1 TO I
1450  LET A0=A0+(2*D(K)*S(K)/(H(K)-X*U(K)))^.5*U(K)/2
1460  NEXT K
1470  RETURN
1480  REM ** SUBROUTINE FOR COMPUTING FIRST DERIVATIVE **
1490  LET D=0
1500  FOR K=1 TO I
1510  LET D=D+(2*D(K)*S(K))^.5*(U(K)^2)/(4*(H(K)-X*U(K))^1.5)
1520  NEXT K
1530  RETURN
1540  REM**SUBROUTINE FOR FINDING MINIMUM H/U**
1550  FOR K=1 TO I
1560  LET R(K)=H(K)/U(K)
1570  NEXT K
1580  LET M=R(1)
1590  FOR K=2 TO I
```

```
1600  IF R(K)<M THEN 1620
1610  GOTO 1630
1620  LET M=R(K)
1630  NEXT K
1640  RETURN
1650  REM **SUBROUTINE 1 FOR FIBONACCI NUMBERS**
1660  LET F0=1
1670  LET F1=1
1680  LET N=1
1690  LET F=F1
1700  IF F>=T THEN 1760
1710  LET F=F1+F0
1720  LET F0=F1
1730  LET F1=F
1740  LET N=N+1
1750  GOTO 1700
1760  RETURN
1770  REM**SUBROUTINE 2 FOR FIBONACCI NUMBERS**
1780  IF J=0 THEN 1810
1790  IF J=1 THEN 1810
1800  GOTO 1830
1810  LET G=1
1820  GOTO 1900
1830  LET G0=1
1840  LET G1=1
1850  FOR K=2 TO J
1860  LET G=G1+G0
1870  LET G0=G1
1880  LET G1=G
1890  NEXT K
1900  RETURN
1910  REM *****ALL SUBROUTINES END HERE*****
1920  DATA 10,1,1,2000,10,1,5,2000
1930  DATA 30,2,2,10000
1940  PRINT
1950  PRINT
1960  PRINT
1970  PRINT 'MODIFIED LATTICE-FIBONACCI SEARCH ALGORITHM'
1980  PRINT '==========================================='
1990  PRINT
2000  PRINT 'NUMBER OF ITEMS = ':I
2010  PRINT 'IMPOSED AGGREGATE INVENTORY = ':A
2020  PRINT 'MAXIMUM ADMISSIBLE ERROR = ':B:'%'
2030  PRINT
2040  IF W2=1 THEN 2090
2050  PRINT
2060  PRINT 'LATTICE SEARCH BY BOLZANO METHOD'
2070  PRINT '==============================='
2080  PRINT
2090  IF W2=2 THEN 2140
2100  PRINT 'LENGTH OF INITIAL BRACKETING INTERVAL = ':L
2110  PRINT 'ACTUAL NUMBER OF ITERATIONS = ':W
2120  PRINT 'ITERATIONS REQUIRED BY PURE FIBONACCI SEARCH = ':N0
```

```
2130  IF W2=1 THEN 2150
2140  PRINT 'NUMBER OF ITERATIONS = ':W1
2150  PRINT
2160  PRINT 'ESTIMATED LAGRANGE MULTIPLIER = ':X
2170  PRINT 'RESULTING AGGREGATE INVENTORY = ':A0
2180  PRINT 'ACTUAL ERROR IN AGGREGATE INVENTORY = ':P:'%'
2190  PRINT
2200  PRINT 'PRODUCTION PLAN'
2210  FOR K=1 TO I
2220  PRINT 'ITEM ':K,Q(K):'UNITS'
2230  NEXT K
2240  GOTO 2330
2250  REM ** PARTITION BRACKETING INTERVAL FOR BOLZANO SEARCH **
2260  LET K=1
2270  LET V=2^K
2280  IF V>=T THEN 2310
2290  LET K=K+1
2300  GOTO 2270
2310  RETURN
2320  REM ** BOLZANO SEARCH **
2330  IF W2=2 THEN 2630
2340  LET W1=0
2350  LET V1=V/2
2360  LET X=E3+V1*(E4-E3)/V
2370  GOSUB 1420
2380  LET W1=W1+1
2390  LET Y=A0-A
2400  LET X0=X
2410  IF Y=0 THEN 1320
2420  IF Y<0 THEN 2450
2430  LET E4=X
2440  GOTO 2460
2450  LET E3=X
2460  IF V1=1 THEN 2490
2470  LET V=V-V1
2480  GOTO 2350
2490  LET X=E3
2500  GOSUB 1420
2510  LET Y1=A0-A
2520  LET Y1=ABS(Y1)
2530  LET X=E4
2540  GOSUB 1420
2550  LET Y2=A0-A
2560  IF Y1<=Y2 THEN 2590
2570  LET X0=E4
2580  GOTO 2600
2590  LET X0=E3
2600  LET W2=2
2610  GOTO 1320
2620  REM ** END OF BOLZANO SEARCH **
2630  END
```