Entropy Optimization in Mathematical Programming

R. Kar
Dept. of Mathematics Bidhan Chandra College,
Rishra, Hooghly, 712248, West Bengal, India

and

S. K. Mazumder
Dept. of Mathematics Bengal Engineering
and Science University, Shibpur, Howrah,
711103, West Bengal, India

Received February 20, 2008, Accepted September 25, 2008

Abstract
The Paper concentrates upon a modified maximum-entropy method of solution of linear and non-linear programming problem transforming it by an equivalent surrogate program.

Keywords and Phrases: Entropy, Optimization, Surrogate, Lagrangian.
1. Introduction

Let us consider a general optimization problem of the following form:

Problem – I

Minimize \( g(x) = g(x_1, x_2, \ldots, x_n) \)

\[ (1.1) \]

\[ \{ x_i, i = 1, 2, \ldots, n \} \]

Subject to the constraints \( d_j(x) \leq 0, j = 1, 2, \ldots, m \)

\[ (1.2) \]

Where \( x \) denotes vector of real, continuous-valued variables \( x_i(i = 1, 2, \ldots, n) \) and the constraint functions \( d_j(j = 1, 2, \ldots, m) \) represent constraint vector \( d \). Problems of type – I with non-linear functions \( g \) have far reaching applications and there are plenty of techniques of solving this type of problems [11, 14]. The numerical solutions of this type of problem are of great importance for engineering design, synthesis and analysis [4]. Numerical solution of this type of problems on the basis of Jaynes maximum-entropy principle [9] is first due to Templeman and Li [17]. The method was modified by Das, Mazumder and De [2] on the basis of kullback minimum cross-entropy principle [10]. Later on Das and Chakrabarti [3] has formulated a modified maximum-entropy principle, which removes some flaws and demerits of the paper of Templeman and Li [17] and that of Das, Mazumder and De [2]. This method enables a powerful algorithm of tackling numerical optimization problems, which are very often of great need in engineering science.

The object of the paper is to show how this modified maximum-entropy algorithm technique can be successfully applied to the numerical solution of the linear and non-linear programming problems. We have attempted to establish our claims with four numerical examples pointing out the advantageous points of the technique.
2. Equivalent Surrogate Programming: Modified

Maximum-Entropy Approach

For clear understanding of the mathematical background of the technique we describe it briefly as follows: The Problem I has an equivalent surrogate form [17].

**Problem – II**

Minimize \( g(x) = (x_1, x_2, \ldots, x_n) \)

\( \{ x_i, i = 1, 2, \ldots, n \} \)

Subject to the single constraint

\[
\sum_{j=1}^{m} \mu_j d_i(x) = 0 \quad (2.1)
\]

Where \( \mu_j (j = 1, 2, \ldots, m) \) are non-negative multipliers known as surrogate multipliers. The surrogate multipliers \( \mu \) may be assumed to be normalized without loss of generality, i.e.

\[
\sum_{j=1}^{m} \mu_j = 1 \quad (2.2)
\]

The solution process is set in a probabilistic context by considering the **Problem I** as posed initially and estimating what level of certainty (probability) should be assigned to the event that each constraint is active at the problem solution. Denoting these probability by \( \mu_j (j = 1, 2, \ldots, m) \) it is known that at least one of the constraints must be active so that (2.2) must hold. The desired solution \( x^* \) of the **Problem I** will here be sought indirectly through a sequence of solutions of the **Problem II**. This approach assumes therefore, that **Problems I and II** are equivalent at
the solution point specially that a set of multipliers $\mu^*$ exists and can be found such that $x^*$ which solves Problem II with $\mu^*$ also solves Problem I.

The Lagrangian of problem II has the form

$$L_{II}(x, \mu, a) = g(x) + \beta \sum_{j=1}^{m} \mu_j d_j(x)$$  \hspace{1cm} (2.3)

Where $\beta$ is the Lagrange multiplier associated with the constraint (2.1) in problem II. An essential condition to be satisfied is that $(x^*, \mu^*, a^*)$ should be a saddle point of the lagrangian $L_{II}$ of the Problem II

$$L_{II}(x, \mu^*, \beta^*) \geq L_{II}(x^*, \mu^*, a^*) \geq L_{II}(x^*, \mu, \beta)$$  \hspace{1cm} (2.4)

The saddle point condition (2.4) may be satisfied, however by iterative means using problem II itself with alternative iterations in $x$ – space and $\mu$ space. A typical scheme is as follows :

An initial set of multipliers $\mu^0$ is chosen and problem II is solved to yield corresponding values of $x^0$. The multipliers are then updated to $\mu^1$ and problem II is solved again to give $x^1$. The process is repeated until the sequence $(\mu^0, x^0), (\mu^1, x^1), \ldots, (\mu^k, x^k)$, converges to the solution of problem II and hence also of problem I, at $(x^*, \mu^*)$. The iteration process should satisfy the inequality [17]

$$\begin{cases} 
\mu_j^{k+1} = \mu_j^k & \text{if } d_j(x^k) = 0 \\
> \mu_j^k & \text{if } d_j(x^k) > 0 \\
< \mu_j^k & \text{if } d_j(x^k) < 0 
\end{cases}$$  \hspace{1cm} (2.5)
The new multipliers $\mu^{k+1}$ must satisfy the normality conditions and the surrogate constraints of problem II. Thus

$$\sum_{j=1}^{m} \mu_j^{k+1} = 1 \quad (2.6)$$

$$\sum_{j=1}^{m} \mu_j^{k+1} d_j(x^{k+1}) = 0 \quad (2.7)$$

For the conditions (2.7) to be satisfied we require values for $d_j(x^{k+1})$, $j = 1, 2, \ldots, m$ which are not known as yet. Therefore, the best current estimates for $d_j(x^{k+1})$ are the values $d_j(x^k)$ which are available and must be used in their place. Consequently (2.7) is modified to

$$\sum_{j=1}^{m} \mu_j^{k+1} d_j(x^k) = \varepsilon \quad (2.8)$$

Where $\varepsilon$ represents the unknown error introduced by using $d_j(x^k)$ in place of $d_j(x^{k+1})$, $j = 1, 2, \ldots, m$. Furthermore, we would expect $\varepsilon$ to approach zero as the sequence of iterations $x_k$, $x_{k+1}$, $\ldots$ approaches to $x^*$.

Now let us write $y_{j,k+1} = (\mu_{j,k+1} - \mu_{j,k}) d_j(x^k)$, $j = 1, 2, \ldots, m$. (2.9)

In view of (2.1) and (2.8) we have then

$$\sum_{j=1}^{m} y_{j,k+1} = \sum_{j=1}^{m} (\mu_{j,k+1} - \mu_{j,k}) d_j(x^k) = \varepsilon$$
Which tends to zero as $k \to \infty$

Since \[ \sum_{j=1}^{m} \mu_j^k d_j(x^k) = 0 \] we thus have then

\[ \sum_{j=1}^{m} y_j^{k+1} = \varepsilon \] (2.10)

\[ \sum_{j=1}^{m} \frac{y_j^{k+1}}{d(x^k)} = 0 \] (2.11)

and \[ \mu_j^{k+1} \geq 0, \quad j = 1, 2, \ldots, m. \] (2.12)

To determine $y_j^{k+1}$ we pose the following problem:

**Problem – III**

Maximize \[ S = - \sum_{j=1}^{m} y_j^{k+1} \ln y_j^{k+1} \] (2.13)

Subject to the constraints \[ \sum_{j=1}^{m} y_j^{k+1} = \varepsilon \] (2.14)

\[ \sum_{j=1}^{m} \frac{y_j^{k+1}}{d(x^k)} = 0 \] (2.15)

and \[ \mu_j^{k+1} \geq 0, \quad j = 1, 2, \ldots, m. \] (2.16)

The solution of problem III gives:

\[ \ln y_j^{k+1} = (\alpha - 1) + \frac{\eta}{d(x^k)} \quad j = 1, 2, \ldots, m. \]
Or \[ y_{j}^{k+1} = B e^{\eta/d_j(x^k)} \] (2.17)

Where \( \eta \) is determined from the equation

\[
\sum_{j=1}^{m} e^{\eta/d_j(x^k)} = 0
\]

(2.18)

Let the value of \( \eta \) be \( \eta^* \) then from (2.17), we have

\[ y_{j}^{k+1} = B e^{\eta^*/d_j(x^k)} \]

So,

\[ \sum_{j=1}^{m} y_{j}^{k+1} = \sum_{j=1}^{m} B e^{\eta^*/d_j(x^k)} = \varepsilon \]

Implying \( B \sum_{j=1}^{m} B e^{\eta^*/d_j(x^k)} = \varepsilon \) (2.19)

Since we expect \( \varepsilon \) gradually approaches to zero as the iteration proceeds (i.e \( \varepsilon \to 0 \) as \( K \to \infty \) ) therefore, equation (2.19) states that \( B \) is a non-negative quantity which infact gradually approaches to zero as the iteration increases.

By virtue of (2.9) the solution of the problem III gives the following assignment of \( \mu_j^{k+1} \):

\[ \mu_j^{k+1} = \mu_j^k + \frac{B e^{\eta/d_j(x^k)}}{d_j(x^k)}, \quad j = 1, 2, \ldots, m \] (2.20)

\[ \mu_j^{k+1} \geq 0 \quad \text{we have} \]

\[ B \leq \frac{-\mu_j^k d_j(x^k)}{e^{\eta/d_j(x^k)}}, \quad j = 1, 2, \ldots, m \] (2.21)
In view of this it should be possible to prescribe procedures for selecting sequence \( \{B\} \) of positive numbers that converges to zero. The above method satisfies the conditions (2.5) absolutely confirming the convergence of \( \mu \) [3].

3. Application to Programs


In the present section we are going to deal with the technique of section 2 in the solution of linear and non-linear programming problems. This technique has been shown to be very successful in generating an iterative procedure in the numerical solution of linear and non-linear programs. The different steps in the iterative procedure are as follows:

a) The first step is to reduce the original program into a surrogate and reduce the number of constraints into a single constraint.

b) The next step is to add an entropy term to the new objective function so that the new objective function becomes a non-linear function (in case of linear program) of its arguments in consistent with the formalism of the section 2.

c) Next is to write down the lagrangian for the surrogate problem introducing a Lagrangian parameter \( \beta \) (say).
d) Maximization of the Lagrangian with respect to the new surrogate variables $y$ (say) leads to the set of equations which determine the lagrangian parameter $\beta$.

e) We start with the computation scheme by setting surrogate multipliers $\mu$ equal so that their total sum be unity. This is consistent with Laplace’s principle of insufficient knowledge—a particular case of Jaynes’ maximum-entropy principle [9].

f) By starting with the initial values $\mu_j^0$, we upgrade their values step by step and find the corresponding values of the surrogate variables $y_i$ and hence original variables $x_i$ ($i = 1, 2, \ldots, n$). The whole iterative process can be arranged in a tabular form for a better representation of the solution. The procedure has been explained more explicitly by four numerical examples presented below.

**Numerical Example 1**:

A firm can produce three types of cloth, say A, B and C. Three kinds of wool are required for it, say, red wool, green wool and blue wool. One unit length of type A cloth needs 2 yards of red wool and 3 yards of blue wool; one unit length of type B cloth needs 3 yards of red wool, 2 yards of green wool and 2 yards of blue wool and one unit of type C cloth needs 5 yards of green wool and 4 yards of blue wool. The firm has only a stock of 8 yards of red wool, 10 yards of green wool and 15 yards of blue wool. It is assumed that the income obtained from the one unit length of type A cloth is Rs. 3.00, of type B cloth is Rs. 5.00 and of type C cloth is Rs. 4.00.

The L.P. model is written as:

Maximize $Z = 3x_1 + 5x_2 + 4x_3$

Subject to constraints:

\[
\begin{align*}
  d_1 &\equiv 2x_1 + 3x_2 - 8 \leq 0 \\
  d_2 &\equiv 2x_2 + 5x_3 - 10 \leq 0 \\
  d_3 &\equiv 3x_1 + 2x_2 + 4x_3 - 15 \leq 0
\end{align*}
\]

(1.a)

and $x_1, x_2, x_3 \geq 0$
The surrogate form of (1.a) is given by

Maximize \( Z = 3x_1 + 5x_2 + 4x_3 - x_1\ln x_1 - x_2\ln x_2 - x_3\ln x_3 \)

Subject to
\[ \mu_1 (2x_1 + 3x_2 - 8) + \mu_2 (2x_2 + 5x_3 - 10) + \mu_3 (3x_1 + 2x_2 + 4x_3 - 15) = 0 \]

Or \( a_1x_1 + a_2x_2 + a_3x_3 - a_4 = 0 \) (1.b)

Where \( a_1 = (2\mu_1 + 3\mu_3) \)
\( a_2 = (3\mu_1 + 2\mu_2 + 2\mu_3) \)
\( a_3 = (5\mu_2 + 4\mu_3) \)
\( a_4 = (8\mu_1 + 10\mu_2 + 15\mu_3) \) (1.c)

Now the lagrangian for the above surrogate problem is

\[ L(X, \beta) = (3x_1 + 5x_2 + 4x_3 - x_1\ln x_1 - x_2\ln x_2 - x_3\ln x_3) + \beta (a_1x_1 + a_2x_2 + a_3x_3 - a_4) \] (1.d)

Now, the maximization requires

\[ \frac{\partial L}{\partial x_1} = 0 \Rightarrow x_1 = e^{(2+a_1)\beta} \] (1.e)

\[ \frac{\partial L}{\partial x_2} = 0 \Rightarrow x_2 = e^{(4+a_2)\beta} \] (1.f)

\[ \frac{\partial L}{\partial x_3} = 0 \Rightarrow x_3 = e^{(3+a_3)\beta} \] (1.g)

Putting \( x_1, x_2, x_3 \), in (1.b) we have

\[ a_1e^{(2+a_1)\beta} + a_2e^{(4+a_2)\beta} + a_3e^{(3+a_3)\beta} - a_4 = 0 \] (1.h)
Which gives the value of $\beta$ for known $a_1, a_2, a_3, a_4$ (determined from (1.c) for given $\mu_1, \mu_2, \mu_3$). We start the process by setting $\mu_1 = \mu_2 = \mu_3 = \frac{1}{3}$. Infact Jaynes maximum-entropy principle’ generates this starting set for $\mu_1, \mu_2, \mu_3$ because in the absence of any others extra information about the problem the least biased assumption that we can make is that all the constraints are equally weighted. Table 1 give the iterative result and is as exact as the three decimal place accuracy.

**TABLE – 1**

<table>
<thead>
<tr>
<th>K</th>
<th>B</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\mu_3$</th>
<th>$\beta$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$d_3$</th>
<th>$\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>$\frac{1}{3}$</td>
<td>1.20233</td>
<td>.99611</td>
<td>3.32083</td>
<td>.54498</td>
<td>3.84653</td>
<td>-</td>
<td>-.67092</td>
<td>3.22755</td>
<td>1.44349</td>
</tr>
<tr>
<td>1</td>
<td>.0008</td>
<td>.33630</td>
<td>.33194</td>
<td>.33174</td>
<td>1.20287</td>
<td>.99379</td>
<td>3.28613</td>
<td>.55285</td>
<td>3.84521</td>
<td>-.66347</td>
<td>3.23494</td>
<td>1.42105</td>
</tr>
<tr>
<td>2</td>
<td>.0004</td>
<td>.33645</td>
<td>.33187</td>
<td>.33166</td>
<td>1.20290</td>
<td>.99367</td>
<td>3.28532</td>
<td>.55325</td>
<td>3.84333</td>
<td>-.66307</td>
<td>3.23530</td>
<td>1.41990</td>
</tr>
<tr>
<td>3</td>
<td>.0001</td>
<td>.33649</td>
<td>.33184</td>
<td>.33163</td>
<td>1.20291</td>
<td>.99364</td>
<td>3.28512</td>
<td>.55335</td>
<td>3.84266</td>
<td>-.66297</td>
<td>3.23539</td>
<td>1.41962</td>
</tr>
<tr>
<td>4</td>
<td>.00007</td>
<td>.33651</td>
<td>.33184</td>
<td>.33163</td>
<td>1.20291</td>
<td>.99362</td>
<td>3.28501</td>
<td>.55339</td>
<td>3.84210</td>
<td>-.66290</td>
<td>3.23546</td>
<td>1.419418</td>
</tr>
</tbody>
</table>

**Numerical Example 2 :**

Let us consider the following optimization problem, which is of significant importance in engineering design [11]. A firm manufactures two types of products of $x_1$ and $x_2$ units. The profit function is given by

$$f(x_1, x_2) = x_1 + 3x_2$$

The problem here to maximize the linear function

Maximize $f(x_1, x_2) = x_1 + 3x_2$  \hspace{1cm} (2.a)

Subject to constraints :

$$3x_1 + 6x_2 \leq 8$$
$$5x_1 + 2x_2 \leq 10, \hspace{0.5cm} x_1, x_2 \geq 0$$ \hspace{1cm} (2.b)
Now first rewrite the L.P.P by introducing a transformation of variables defined by

\[ y_1 = \frac{x_1}{2} \quad \text{and} \quad y_2 = \frac{x_2}{5} \]

We use this transformation in order to make one of the constraints of (2.b) in its normality from which in fact is consistent with the method developed in this paper. Then the equation (2.a) and (2.b) transformed to

Maximize \[ f(y_1, y_2) = 2y_1 + 15y_2 \]
Subject to
\[ \begin{align*}
  d_1 &\equiv 3y_1 + 15y_2 - 4 \leq 0 \\
  d_2 &\equiv y_1 + y_2 - 1 \leq 0 \\
  y_1, y_2 &\geq 0
\end{align*} \] (2.c)

The surrogate form of (2.c) is given by

Maximize \[ \tilde{f}(y_1, y_2) = 2y_1 + 15y_2 - y_1 \ln y_1 - y_2 \ln y_2 \]
Subject to \[ \mu_1 (3y_1 + 15y_2 - 4) + \mu_2 (y_1 + y_2 - 1) = 0 \]
Or \[ a_1 y_1 + a_2 y_2 - a_3 = 0 \] (2.d)

Where
\[ \begin{align*}
  a_1 &= (3\mu_1 + \mu_2) \\
  a_2 &= (15\mu_1 + \mu_2) \\
  a_3 &= (4\mu_1 + \mu_2)
\end{align*} \] (2.e)

Following the same argument as in Numerical Example 1, Table 2 give the iterative results and is as exact as the three decimal place accuracy.

\[ \text{TABLE – 2} \]

<table>
<thead>
<tr>
<th>K</th>
<th>B</th>
<th>(\mu_1)</th>
<th>(\mu_2)</th>
<th>(\beta)</th>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(d_1)</th>
<th>(d_2)</th>
<th>(\eta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>–</td>
<td>(\frac{1}{2})</td>
<td>(\frac{1}{2})</td>
<td>-1.90160</td>
<td>.606149E-01</td>
<td>.29734</td>
<td>.64230</td>
<td>-.64203</td>
<td>-0.173194E-13</td>
</tr>
</tbody>
</table>
Numerical Example 3:

In this example we study with the non-linear programming problem.

Minimize \( f(x_1, x_2) = -4x_1 + x_1^2 - 2x_1x_2 + 2x_2^2 \)

Subject to the constraints:

\[
\begin{align*}
  d_1 &= 2x_1 + x_2 - 6 \leq 0 \\
  d_2 &= x_1 - 4x_2 \leq 0 , \quad x_1, x_2 \geq 0
\end{align*}
\]

The surrogate form of (3.a) is given by

Minimize \( \tilde{f}(x_1, x_2) = -4x_1 + x_1^2 - 2x_1x_2 + 2x_2^2 + x_1\ln x_1 + x_2\ln x_2 \)

Subject to \( \mu_1 (2x_1 + x_2 - 6) + \mu_2 (x_1 - 4x_2) = 0 \)

Or, \( a_1 x_1 + a_2 x_2 - a_3 = 0 \)

Where

\[
\begin{align*}
  a_1 &= (2\mu_1 + \mu_2) \\
  a_2 &= (\mu_1 - 4\mu_2) \\
  a_3 &= -6\mu_1
\end{align*}
\]

Now the Lagrangian for the above surrogate problem is

\[ L(X, \beta) = -4x_1 + x_1^2 - 2x_1x_2 + 2x_2^2 + x_1\ln x_1 + x_2\ln x_2 + \beta (a_1 x_1 + a_2 x_2 - a_3) \]

Now the maximization requires

\[
\frac{\partial L}{\partial x_1} = 0 \implies 2x_1 - 2x_2 + \ln x_1 + \beta a_1 - 3 = 0 \quad (3.d)
\]

\[
\frac{\partial L}{\partial x_2} = 0 \implies -2x_1 + 4x_2 + \ln x_2 + \beta a_2 + 1 = 0 \quad (3.e)
\]
Now we follow the same argument as in Numerical Example 1, Table 3 give the iterative results and is as exact as the three decimal place accuracy.

### Table 3

<table>
<thead>
<tr>
<th>K</th>
<th>B</th>
<th>$\mu_1$</th>
<th>$\mu_2$</th>
<th>$\beta$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1$</th>
<th>$d_2$</th>
<th>$\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>$\frac{1}{2}$</td>
<td>$\frac{1}{2}$</td>
<td>0</td>
<td>2.86934</td>
<td>1.14978</td>
<td>.88847</td>
<td>1.00005</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>.007</td>
<td>.50787</td>
<td>.50700</td>
<td>0</td>
<td>2.86846</td>
<td>1.14942</td>
<td>.88635</td>
<td>1.00008</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>.001</td>
<td>.52601</td>
<td>.50900</td>
<td>0</td>
<td>2.86834</td>
<td>1.14931</td>
<td>.88605</td>
<td>1.00009</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>.0005</td>
<td>.50957</td>
<td>.50850</td>
<td>0</td>
<td>2.86830</td>
<td>1.14922</td>
<td>.88590</td>
<td>1.00010</td>
<td>0</td>
</tr>
</tbody>
</table>

**Numerical Example 4:**

Here also we study with the non-linear programming problem and its constraints also non-linear.

Minimize $f(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 5)^2$

Subject to the constraints:

\[
\begin{align*}
    d_1 &\equiv -x_1^2 + x_2 - 4 \leq 0 \\
    d_2 &\equiv -(x_1 - 2)^2 + x_2 - 3 \leq 0, \quad x_1, x_2 \geq 0
\end{align*}
\]

The surrogate form of (4.a) is given by

Minimize $\tilde{f}(x_1, x_2) = (x_1 - 1)^2 + (x_2 - 5)^2 + x_1 \ln x_1 + x_2 \ln x_2$

Subject to $\mu_1 ( - x_1^2 + x_2 - 4) + \mu_2 \{ - (x_1 - 2)^2 + x_2 - 3 \} = 0$

Now the Lagrangian for the above surrogate problem is
\[ L(X, \beta) = (x_1 - 1)^2 + (x_2 - 5)^2 + x_1 \ln x_1 + x_2 \ln x_2 + \beta [\mu_1 (x_1^2 + x_2 - 4) + \mu_2 (x_1 - 2)^2 + x_2 - 3] \]

Now the maximization requires

\[ \frac{\partial L}{\partial x_1} = 0 \Rightarrow 2(1 - \beta \mu_1 - \beta \mu_2) x_1 + 4 \beta \mu_2 + \ln x_1 - 1 = 0 \] \hspace{1cm} (4.c)

\[ \frac{\partial L}{\partial x_2} = 0 \Rightarrow 2 x_2 + \ln x_2 + \beta (\mu_1 + \mu_2) - 9 = 0 \] \hspace{1cm} (4.d)

Now we follow the same argument as in Numerical Example 1, Table 4 give the iterative results and is as exact as the three decimal place.

**TABLE – 4**

<table>
<thead>
<tr>
<th>K</th>
<th>B</th>
<th>(\mu_1)</th>
<th>(\mu_2)</th>
<th>(\beta)</th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(d_1)</th>
<th>(d_2)</th>
<th>(\eta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>–</td>
<td>(\frac{1}{2})</td>
<td>(\frac{1}{2})</td>
<td>2.23897</td>
<td>1.59097</td>
<td>2.85585</td>
<td>1.38703</td>
<td>1.00000</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>.0009</td>
<td>.500648</td>
<td>.500899</td>
<td>2.23894</td>
<td>1.59096</td>
<td>2.85584</td>
<td>1.38700</td>
<td>1.00000</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>.0003</td>
<td>.500865</td>
<td>.501199</td>
<td>2.23652</td>
<td>1.59089</td>
<td>2.85539</td>
<td>1.38635</td>
<td>1.00010</td>
<td>0</td>
</tr>
</tbody>
</table>

**4. Conclusion**

In this paper we have analysed the applicability of the modified maximum-entropy principle in the numerical solution of linear and non-linear programming problems. Infact, it may be justified to conclude the paper by saying that this metamorphosis to the surrogate mode has diminished the number of hindrances into a single one and, furthermore, this has lessened the manifold numerical complications of the problem undoubtedly to a large measure.
References


