NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.
THE LINEAR-LOGARITHMIC PROGRAMMING PROBLEM

R. J. Clasei

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND
THE LINEAR-LOGARITHMIC PROGRAMMING PROBLEM

R. J. Clasen

This research is sponsored by the United States Air Force under Project RAND—contract No. AF 49(638)-700 monitored by the Directorate of Development Planning, Deputy Chief of Staff, Research and Development, Hq USAF. Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the United States Air Force.
The linear-logarithmic programming method documented in this Memorandum was developed to handle chemical equilibrium problems in computer modeling. It forms a part of the continuing program of research at The RAND Corporation in medicine and biology, and was motivated primarily in conjunction with the work of J. C. DeHaven and E. C. DeLand in modeling biological and physiological processes.

The particular method may also be useful in other modeling applications in which linear and logarithmic terms are present.
SUMMARY

In this Memorandum is developed the algebraic solution of the linear-logarithmic programming problem, derived by means of Lagrange multipliers. Then, two numerical methods for solving the problem are given, one of which is a generalization of a method previously used [3] to solve the chemical equilibrium problem. Convergence has not been proven for either of these methods; however, a number of large chemical equilibrium problems have been solved using one or both of the methods.
THE LINEAR-LOGARITHMIC PROGRAMMING PROBLEM

We wish to consider the problem of minimizing

\[ \phi = \sum_{j=1}^{N} x_j (c_j + d_j \log x_j) \]  

while satisfying the constraints

\[ \sum_{j=1}^{N} a_{ij} x_j - b_i = 0 ; \quad i = 1, 2, 3, \ldots, M \]  

where \( a_{ij}, b_i, c_j, d_j \), are constants and \( x_j \) are unknowns.

If all \( d_j \) are zero, we have a linear programming problem--the case that we consider here is when none of the \( d_j \) are zero. Using the Lagrange multiplier method of solving the problem, we set

\[ L = \phi + \sum_{i=1}^{M} \pi_i \left( \sum_{j=1}^{N} a_{ij} x_j - b_i \right). \]

A local extremum is obtained if

\[ \frac{\partial L}{\partial x_j} = 0 ; \quad j = 1, 2, 3, \ldots, N \]

or

\[ c_j + d_j \log x_j + d_j + \sum_{i=1}^{M} \pi_i a_{ij} = 0 ; \quad j = 1, 2, 3, \ldots, N. \]
Solving (3) we have

\[
\log x_j = \frac{-c_j - d_j - \sum_{i=1}^{M} n_i a_{ij}}{d_j}; \\
\text{or,} \\
\begin{align*}
x_j &= \exp \left\{ \frac{-c_j - d_j - \sum_{i=1}^{M} n_i a_{ij}}{d_j} \right\}; \\
\text{for } j &= 1, 2, 3, \ldots, N.
\end{align*}
\] (4)

Note that for (5) to be a solution of the problem, we must have all \( x_j > 0 \). Let us thus assume that all \( x_j > 0 \). The problem then reduces to that of determining \( M \) \( \pi_i \)'s so that the \( x_j \) from (5) satisfy (2). Equivalently, the \( M + N \) equations in (2) and (5) must be satisfied simultaneously by proper choice of the \( N \) unknown \( x_j \) and the \( M \) unknown \( \pi_i \).

Since (5) is non-linear, we cannot solve it directly, but must resort to approximation schemes. We now consider two methods of approximating the solution.

**METHOD 1 (Delta-X Method)**

Suppose we have an estimate of the \( x_j \) which may or may not satisfy (2). Let us denote the estimate by \( y_j \). Then,
the Taylor expansion of \( \log x_j \) about \( y_j \) is

\[
\log x_j = \log y_j + \frac{1}{y_j} (x_j - y_j) + \text{higher order terms.}
\]

Dropping the higher order terms, we have

\[
\log x_j \approx \log y_j + \frac{x_j}{y_j} - 1. \tag{6}
\]

Substituting (6) in (4), we have

\[
\begin{align*}
\log y_j + \frac{x_j}{y_j} - 1 & \approx -c_j - d_j - \sum_{i=1}^{M} a_{ij} \\
\end{align*}
\]

or,

\[
\begin{align*}
x_j = -y_j \left[ \log y_j + \frac{c_j + \sum_{i=1}^{M} a_{ij}}{d_j} \right]. \tag{7}
\end{align*}
\]

Substituting (7) in (2), we have

\[
\begin{align*}
\sum_{t=1}^{M} \left( \sum_{j=1}^{N} \frac{a_{ij} a_{lj} y_j}{d_j} \right)^{\pi_{tl}} = -b_1 - \sum_{j=1}^{N} a_{ij} y_j \left( \log y_j + \frac{c_j}{d_j} \right); \\
i = 1, 2, 3, \ldots, M.
\end{align*}
\]
Letting

\[ r_{il} = \sum_{j=1}^{N} \frac{a_{ij} a_{lj} y_j}{d_j} \]

\[ S_i = -b_i - \sum_{j=1}^{N} a_{ij} y_j \left( \log y_i + \frac{c_i}{d_j} \right), \]

we have

\[ \sum_{l=1}^{M} r_{il} \pi_l = S_i; \quad i = 1, 2, 3, \ldots, M. \quad (8) \]

To solve the problem we do the following: Solve (8) for the M unknown \( \pi_l \) and substitute in (7) to get \( x_j \). Then set \( y_j = x_j \) and repeat the process. When the change in \( x_j \) becomes negligible, we have reached a solution.

**METHOD 2** (Delta-\( \pi \) Method)

In this method we assume that we have an approximation to the M \( \pi_i \)'s. Substituting these \( \pi_i \) into (5) we get values for \( x_j \). Now, rewrite (2) as

\[ g_i = \sum_{j=1}^{N} a_{ij} x_j - b_i; \quad i = 1, 2, 3, \ldots, M. \quad (9) \]

Substituting the above \( x_j \) into (9), we calculate the \( g_i \). If \( g_i = 0 \), all \( i \), we have a solution.
Since, in general, this will not be the case, we wish to change $\pi_i$ so that $g_i \rightarrow 0$. With this in mind, we compute,

$$\frac{\partial g_t}{\partial \pi_i} = \frac{\partial}{\partial \pi_i} \left[ \sum_{j=1}^N a_{tj} x_j - b_t \right] = \sum_{j=1}^N a_{tj} \frac{\partial x_i}{\partial \pi_i}$$

$$= \sum_{j=1}^N a_{tj} \frac{\partial}{\partial \pi_i} \left[ \exp \left( -c_j - d_j - \sum_{k=1}^M \pi_i a_{kj} \right) \right]$$

$$= - \sum_{j=1}^N \frac{a_{tj} x_j a_{jj}}{d_j} a_{jj}.$$

(10)

Denoting the new values of $\pi_i$ by $\pi_i'$ and $p_i = \pi_i' - \pi_i$, we may write

$$dg_t = \sum_{i=1}^M \frac{\partial g_t}{\partial \pi_i} dp_i.$$

(11)

Now, approximating $dg_t$ by $-g_t$ and letting

$$r_{ti} = - \frac{\partial g_t}{\partial \pi_i} = \sum_{j=1}^N \frac{a_{tj} x_j a_{jj}}{d_j}$$

...
we have

$$\sum_{i=1}^{M} r_{ii} P_i = g_i; \quad i = 1, \ldots, M.$$  \hspace{1cm} (12)

Here we have $M$ unknown $P_i$ and $M$ equations. The algorithm then is as follows: Substitute the $\pi_i$ into (5) to get $x_j$. Substitute these $x_j$ into (9) to get $g_i$. If these $g_i$ are small, we are done; otherwise, compute $r_{ii}$ and solve (12) for $P_i$. Then change $\pi_i$ to $\pi_i + P_i$ and repeat the process.

THE CHEMICAL EQUILIBRIUM PROBLEM

In the chemical equilibrium problem [1], the variables $x_j$ are partitioned into a number of "compartments." For simplicity, let us say that $x_1, x_2, \ldots, x_{n_1}$ are in compartment 1; $x_{n_1+1}, x_{n_1+2}, \ldots, x_{n_2}$ are in compartment 2, \ldots; $x_{n_p-1+1}, x_{n_p-2+2}, \ldots, x_{n_p}$ are in compartment $p$, where $n_p = n$.

Define $\overline{x}_k = \sum_{j=n_{k-1}+1}^{n_k} x_j$, so $\overline{x}_k$ is the sum of the $x_j$'s in compartment $k$. Define $\sigma_j$ so that if $j$ is in the $k^{th}$ compartment, $\sigma_j = \overline{x}_k$. Then the chemical equilibrium problem is to minimize

...
\[ \phi = \sum_{j=1}^{n} x_j \left( c_j + \log \frac{x_j}{\sigma_j} \right) \]  
(13)

while satisfying

\[ \sum_{j=1}^{n} a_{ij} x_j - b_i = 0; \quad i = 1, 2, 3, \ldots m. \]  
(14)

To get (13) in the form (1), we add \( p \) equations,

\[ \sum_{j=n_{k-1}+1}^{n_k} x_j - \bar{x}_k = 0, \text{ where} \]

\( \bar{x}_k \) is now the \( (n+k)^{th} \) variable and we have a total of \( n + p = N \) variables and \( m + p = M \) equations. Rewriting (13),

\[ \phi = \sum_{j=1}^{n} x_j \left( c_j + \log \frac{x_j}{\sigma_j} \right) \]

\[ = \sum_{j=1}^{n} x_j (c_j + \log x_j) - \sum_{j=1}^{n} x_j \log \sigma_j \]

\[ = \sum_{j=1}^{n} x_j (c_j + \log x_j) - \sum_{j=n+1}^{N} x_j \log x_j. \]  
(15)
Thus, (15) takes the form (1) with $d_j = \pm 1$. Now either Method 1 or Method 2 may be applied to solve the problem. Method 1 turns out to be essentially the method used by White, et al., [3] to solve the problem.

**CONVERGENCE**

The methods as given above do not, in general, converge. It is necessary that all $x_j \geq 0$ for either of the methods to work. Furthermore, no $x_j$ may get too close to zero enroute to the solution. At present, we are conducting experiments with the chemical equilibrium model to find ways of assuring convergence. At this point we can only indicate possible areas of difficulty.

In Method 2, if our $\pi_i$ guesses are not good, the $P_i$ computed by (12) may be large enough so that when the $x_j$ are computed, some $x_j$ is smaller than the smallest number available in the computer and this will cause that $x_j$ to be taken as zero. This may be somewhat alleviated by computing $P_{\text{max}} = \max \{P_i\}$ and then dividing $P_i$ by some multiple of $P_{\text{max}}$. This prevents $P_i$ from exceeding some arbitrary value, and hence, prevents $x_j$ from going to zero.

In Method 1, the computed difference $x_j - y_j$ may be large enough (in absolute value) so that some $x_j$ becomes negative. Here again, we may attenuate the distance that we move in one iteration by dividing $x_j - y_j$ by some number that exceeds unity. In either case, there is no assurance
that we won't oscillate around a possible solution. Our experience thus far has indicated that Method 1 may be best for starting the solution and that Method 2 works better when we are close to the solution.
REFERENCES

