<table>
<thead>
<tr>
<th>Title</th>
<th>Smuggling Method - A Hyper-dynamical Model for Solving Problems of Linear Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Sugita, Motoyosi</td>
</tr>
<tr>
<td>Citation</td>
<td>Hitotsubashi journal of arts and sciences, 1(1): 88-90</td>
</tr>
<tr>
<td>Issue Date</td>
<td>1960-12</td>
</tr>
<tr>
<td>Type</td>
<td>Departmental Bulletin Paper</td>
</tr>
<tr>
<td>Text Version</td>
<td>publisher</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://doi.org/10.15057/4739">http://doi.org/10.15057/4739</a></td>
</tr>
</tbody>
</table>
SMUGGLING METHOD
—A Hyper-dynamical Model for Solving Problems of Linear Programming—

By MOTOYOSI SUGITA

Professor of Physics

I. Introduction

An extension of the Pyne's method\(^1\) is proposed on purpose to solve the problems of linear programming by using digital computer instead of analogue computer.

Let \(Z\) be the objective function, where \(x_1, x_2, \ldots, x_n\) are variables subjected to the restrictions (2) and \(g_k\)'s, \(a_{jk}\)'s and \(b_k\)'s are constants, then

\[
Z = \sum_{k=1}^{n} g_k x_k, \quad (k : 1, 2, \ldots, n) \tag{1}
\]

\[
\sum_{k=1}^{n} a_{jk} x_k \leq b_j, \quad (j : 1, 2, \ldots, m) \tag{2}
\]

\(x_k \geq 0\)

The problem of seeking the maximum of \(Z\) is reduced to getting the solution of the following differential equations, (3):

\[
dx_k/dt = g_k + \sum_{i} a_{ik} (b_i - \sum_{j} a_{ij} x_j) l_{ik}, \quad (k : 1, 2, \ldots, n) \tag{3}
\]

where

\[
l_{ik} = \frac{a_{ik}}{\sqrt{\sum a_{ik}^2}} \tag{4}
\]

is the generalized direction cosign of the normal of the \(i\)-th hyper-plane in \(n\) dimensional euclidean space and \(k_i\)'s are constants under the conditions

\[
k_i = 0, \quad \text{when} \quad b_i - \sum a_{ij} x_j \geq 0 \tag{5}
\]

\[
k_i > 0, \quad \text{when} \quad b_i - \sum a_{ij} x_j < 0
\]

Here, let us rely upon the analogy of mechanics and consider \(Z\) the potential in the \(n\)-dimensional space. \(k_i(b_i - \sum a_{ij} x_j)\) of the second term of the right of (3) corresponds to the restricting force due to the \(i\)-th boundary hyper-surface, when the objective point\(^{1}\) tres passing the \(i\)-th surface goes into the hinter space or the forbidden region. The force is considered acting perpendicular to the \(i\)-th hyper-plane so as to expel the "violating point" out of the restricted region, so that we multiply it with the generalized direction cosign \(l_{ik}\) of (4).

Let us consider hyper-dynamic equation of motion under these generalized forces, which can be given by

\[
m \ddot{x}_k + \gamma^{-1} \frac{dx_k}{dt} = g_k + \sum_{i} k_i (b_i - \sum a_{ij} x_j) l_{ik}, \quad (3')
\]

where \(m\) is the generalized mass of the objective point, which is very small and considered nearly equal to zero. \(\gamma^{-1}\) is the frictional constant for the movement of that point in

---

\(^1\) Insley E. Pyne. Communication and Electronics, May, p. 139, (1956).
this space. If we put \( m = 0 \) and \( \gamma^{-1} = 1 \), we can get (3).

From the analogy of mechanical motion, if \( k_i \)'s are small the objective point trespassing the hyper-planes \( (2) \), goes into the forbidden region and finally may reach the point \( C \) at which

\[
\frac{dx_k}{dt} = 0 \quad x_k = x_k^c
\]  

(6)

In practice we take these values of \( k_i \)'s very small to evade discontinuous change of motion at the contact with the hyper-planes.

At this point \( C \) let us take new values for \( k_i \)'s, which are very large instead of very small. Then the point must be expelled out of the hinder space like the smuggler in a forbidden region which is driven out by authorities. The driving power corresponds to the large values of \( k_i \)'s. The driven point may reach to \( O \), a vertex of the hyper-polyhedron determined by \( (2) \). The practice of computation will be considered in II.

II. The Practice of Computation

Instead of considering the motion of the objective point under the forces depending on constant \( k_i \)'s, let us consider the motion under the forces depending on variable \( k_i \)'s, which is given by

\[
k_i = k(t-t_i)^n
\]  

(7)

where \( t_i \) is the time at which the point is on the \( i \)-th boundary surface, and

\[
k = 0, \quad \text{when} \quad t \leq t_i
\]

\[
k = \text{const} > 0, \quad \text{when} \quad t > t_i
\]  

(8)

\( n \) is a positive value which is taken adequately so that the effect of finite difference \( \Delta t \) on the one hand, can be negligible and on the other \( k_i \) is very large when \( t \) is sufficiently larger than \( t_i \). Then, solving (3) numerically we can get the terminal value \( x_k^c \). The practice of computation may be done using the Runge-Kutta method. The programming of digital computer will be considered in the future.

The motion of the objective point is rather simple. We can see this circumstance if we take the differential equation of \( p_i \) into account, which is introduced by

\[
p_i = \sum a_{ij} x_j - b_i
\]  

(9)

Then we have

\[
t = t_i \quad \text{at} \quad p_i = 0
\]

and from (5) and (8)

\[
k = 0, \quad \text{when} \quad p_i \leq 0
\]

\[
k = \text{const} > 0, \quad \text{when} \quad p_i > 0
\]  

(8')

From (9) we have

\[
dp_i/dt = \sum a_{ij} dx_j/dt = \sum a_{ij} g_j - \Sigma_i k_i p_i \Sigma_j a_{ij} \delta_i = G_i - \Sigma_i k(t-t_i)^n A_i p_i
\]  

(10)

where

\[
G_i = \sum a_{ij} g_j
\]

\[
A_i = \sum \delta_i a_{ij}
\]

Now, let us consider another equation having constant \( k_i = k \) in comparison with (10), i.e.

\[
dp_i/dt = G_i - k \Sigma_i A_i p_i
\]  

(10')

the terminal value of which is \( p_i^c \) corresponding to \( x_k^c \). If \( k \) is sufficiently large, then
$p_i$ must be very small and the solution of (10)' will tend to this value in a simple manner. In the time interval in which $k(t-t_i)$'s are smaller than the value of $k$, the values of $p_i$'s, which are the solution of (10), are larger than those of (10'). Therefore, if we take infinitely large value for $k$, we can see that the values of $p_i$'s are positive in this time interval.

While, if we consider the following equation

$$\frac{dp_i}{dt} = G_i - k(t-t_i)^n A_{ik} p_i$$

which is equivalent to the case of $A_{ik}=0$ in (10) when $i\neq k$.

From (10) and (10'') we can see

$$p_i < p_i$$

Therefore, we can say that the solution of (10), $p_i$, will tend towards zero in a simple way and the objective point will behave in the forbidden region and tends asymptotically to go to the optimum point nearly along the boundary surfaces.

III. Conclusion

The programming for digital computer along the line of our smuggling method is now in consideration. By this method we take at first very small value for $K$ for the benefit of digital computation assuming the variation of $K$ in time as is given by (7) and then try to obtain the values of $x_k$'s integrating (3) numerically. The programming by our method seems much simpler than that of the simplex method. The extension of the Pyne's method using analogue computer is tried by Prof. Jinbo et al of Meiji University. They are intending to apply this method to solve the problem of non-linear programming. The smuggling method here considered also can be applied to such non-linear problems using digital computer.

---