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 conputer.

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

$$Z=\Sigma_k g_k x_k, \qquad (k:1, 2, \ldots\ldots, n) \tag{1}$$

$$\left.\begin{array}{ll}\Sigma_k a_{jk} x_k \leqq b_j & (j:1, 2, \ldots\ldots, m) \\ x_k \geqq 0 & \end{array}\right\} \tag{2}$$

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

$$dx_k/dt=g_k+\Sigma_i k_i(b_i-\Sigma_j a_{ij} x_j) l_{ik}, \qquad (k:1, 2, \ldots\ldots, n) \tag{3}$$

where

$$l_{ik}=\frac{a_{ik}}{\sqrt{\Sigma_k a_{ik}{}^2}} \tag{4}$$

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

$$\left.\begin{array}{lll}k_i=0, & \text{when} & b_i-\Sigma_j a_{ij} x_j \geqq 0 \\ k_i>0, & \text{when} & b_i-\Sigma_j a_{ij} x_j<0\end{array}\right\} \tag{5}$$

Here, let us rely upon the analogy of mechanics and consider Z the potential in the n-dimensional space. $k_i(b_i-\Sigma_j a_{ij} x_j)$ 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

$$m\frac{d^2 x_k}{dt^2}+\gamma^{-1}\frac{dx_k}{dt}=g_k+\Sigma_i k_i(b_i-\Sigma_i a_{ij} x_j) l_{ik}, \tag{3'}$$

where m is the generalized mass of the objective point, which is very small and considered nearly equal to zero. γ^{-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 $m \fallingdotseq 0$ 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

$$(dx_k/dt)_c=0 \qquad x_k=x_k^c \tag{6}$$

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 hinter 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

$$k_i=k(t-t_i)^n \tag{7}$$

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

$$\left.\begin{array}{ll} k=0, & \text{when } t \leqq t_i \\ k=const>0, & \text{when } t>t_i \end{array}\right\} \tag{8}$$

n is a positive value which is taken adequately so that the effect of finite difference Δt 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 Rung-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

$$p_i=\Sigma_j a_{ij}x_j-b_i. \tag{9}$$

Then we have

$$t=t_1 \qquad \text{at } p_i=0$$

and from (5) and (8)

$$\left.\begin{array}{lll} k=0, & \text{when} & p_i \leqq 0 \\ k=const>0, & \text{when} & p_i>0 \end{array}\right\} \tag{8'}$$

From (9) we have

$$dp_i/dt=\Sigma_j a_{ij}dx_j/dt=\Sigma_j a_{ij}g_j-\Sigma_l k_l p_l \Sigma_j a_{ij}l_{lj}=G_i-\Sigma_l k(t-t_1)^n A_{il}p_l \tag{10}$$

where

$$G_i=\Sigma_j a_{ij}g_j$$
$$A_{il}=\Sigma_j a_{ij}l_{lj}$$

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

$$dp_i/dt=G_i-k\Sigma_l A_{il}p_l \tag{10'}$$

the terminal value of which is p_i^c corresponding to x_k^c. If k is sufficiently large, then

p_i^c must be very small and the solution of (10)$'$ will tend to this value in a simple manner. In the time interval in which $k(t-t_i)^n$'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

$$dp_i'/dt=G_i-k(t-t_i)^n A_{ij}p_i'$$

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

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

$$p_j<p_j'$$

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^0'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.[2] 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.

[2] S. Jinbo, Y. Kuroda, Y. Ogawa, and S. Imura, Annual Report of the Institute of Sciences and Technologies, Meiji University Vol. I, 1 (1959).