Journal of the
Operations Research
Society of Japan

VOLUME 2 *October, 1958* NUMBER 1

ON THE COMPUTATIONAL SOLUTION OF DYNAMIC PROGRAMMING PROCESSES, A BOTTLENECK PROCESSES ARISING IN THE STUDY OF INTERDEPENDENT INDUSTRIES

RICHARD BELLMAN and STUART DREYFUS

The Rand Corporation, Santa Monica, California
(*Received May 9, 1958*)

INTRODUCIEON

In (1) and (2), using the functional equation technique of dynamic programming, we discussed a mathematical model associated with the economic question of the most efficient utilization of a complex of interdependent industries.

In this paper, we wish to discuss the numerical solution of a particular class of problems of this general type. As we shall show, a simple device permits us to reduce overtly the computation.

A "BOTTLENECK" PROCESS

Let us assume that we have two interdependent industries, the "auto" industry and "steel" industry, and that the state of each industry at any particular time may be completely specified by two quantities, the stockpile of raw material required for production, and the maximum production capacity. To simplify the problem for this initial computation, we shall assume that the auto capacity is unbounded. The purpose of the process will be taken to produce as many auto as possible over a time-period T.

The process is assumed to be discrete, with allocations made only at times $t=0, 1, 2, \cdots, T-1$. At any particular time, n, let

$$\begin{aligned} x_s(n) &= \text{amount of steel in the steel stockpile} \\ x_m(n) &= \text{capacity of steel mills.} \end{aligned} \tag{1}$$

At each time, n, the steel in the stockpile may be used for either of these purposes, to produce additional steel using the existing steel capacity, to increase the steel capacity, or to produce auto using the existing auto capacity.

Let us write,

$$x_s(n) = z_s(n) + z_m(n) + z_a(n), \tag{2}$$

where

a. $z_s(n)$ = the quantity of steel used to produce an additional steel,

b. $z_m(n)$ = the quantity of steel used to increase steel capacity, (3)

c. $z_a(n)$ = the quantity of steel used to produce autos,

Let us impose the following constraints,

$$\begin{aligned} &\text{a.} \quad z_a(n) \leq a_1 x_s(n), \qquad 0 < a_1 < 1 \\ &\text{b.} \quad z_s(n) \leq x_m(n). \end{aligned} \tag{4}$$

The first constraint says that it is not possible to use more than a fixed percentage of steel for auto production over any stage, k to $k+1$, while the second asserts that there is no point to allocating more steel to the steel mills than the maximum capacity.

Let us now see how the state of the system is affected by the allocations, assuming linearity,

$$\text{a.} \quad x_s(n+1) = a_3 z_s(n), \; a_3 > 1,$$
$$\text{b.} \quad x_m(n+1) = x_m(n) + a_4 z_m(n), \; a_4 > 0 \qquad (5)$$

Finally, let us assume that the quantity of autos produced in a stage is $z_a(n)$.

It is required to choose the quantities $z_s(n)$, $z_m(n)$ and $z_a(n)$, for $n=0, 1, 2, \cdots, T-1$, so as to maximize the total quantity of autos produced over the period $[0, T]$, given the initial quantities,

$$c_1 = x_s(0),$$
$$c_2 = x_m(0). \qquad (6)$$

LINEAR PROGRAMMING APPROACH

The problem reduces to that of maximizing the linear form

$$L(z) = \sum_{n=0}^{T-1} z_a(n), \qquad (1)$$

subject to the constraints of (2), (4) and (5), a problem within the domain of linear programming.

Let us, however, count variables, assuming that we are interested in a 30-stage process. Taking three unknown at each stage, we have a problem involving 90 variables, subject to 120 relations. Although this is not a formidable problem in terms of the simplex method, it is a sizable problem. If we wish to determine the dependence of the solution upon c_1 and c_2, the initial parameters, this method is unwieldly, since it requires a separate computation for each value of c_1 and c_2.

In its place, we wish to present a method which yields the solution for all positive c_1 and c_2, by means of the computation of a sequence of one-dimensional functions.

DYNAMIC PROGRAMMING APPROACH

Let us define, for $N=1, 2, \cdots$,

$f_n(c_1, c_2)$ = total autos production over n stages, starting with initial quantities c_1 and c_2, and using an optimal policy. (1)

Clearly

$$f_1(c_1, c_2) = a_1 c_1. \tag{2}$$

Using the principle of optimality, we have

$$f_R(c_1, c_2) = \max_{\{z\}} [z_a + f_{R-1}(a_3 z_s, c_2 + a_4 z_m)] \tag{3}$$

for $R=2, 3, \cdots, N$, where the maximiztion is over the region in z-space defined by

$$\begin{aligned} &\text{(a)} \quad z_a, z_s, z_m \geq 0, \\ &\text{(b)} \quad z_a + z_s + z_m = c_1, \\ &\text{(b)} \quad z_a \leq a_1 c_1, \\ &\text{(d)} \quad z_s \leq c_2. \end{aligned} \tag{4}$$

In the next section we shall discuss the numerical determination of the sequence $\{f_R(c_1, c_2)\}$ for $R=1, 2, \cdots$, and $c_1, c_2 \geq 0$.

REDUCTION TO A SEARCH OF VERTICES

The region in the (z_a, z_s, z_m)-plane determined by those inequalities has the form of figure 1. These vertices have the following significance in terms of the process:

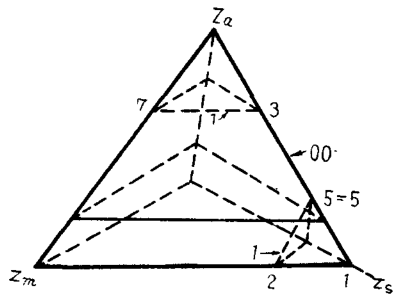

Fig. 1.

1. Where mill capacity does not represent a constraint, allocate all available steel toward the production of more steel.

2. Allocate as much steel as possible towards the production of steel and when the mill capacity is met, allocate the rest toward increasing mill capacity,

3. The allowable percent of steel stockpile is assigned to auto production, the remaining to steel production.

4. The vertex occurs when mill capacity is low. It represents an

allocation to auto production up to the allowable percentage, and an allocation of remaining steel to expanding mill capacity.

5. Allocate as much steel as allowable to steel production and the remainder to auto production.

6. Allocate all of the current steel stockpile toward the expansion of mill capacity.

7. Produce as many autos as possible and then expand mill capacity with the remaining steel.

Obviously all of these conditions do not occur at once, but depend on steel capacity and steel stockpile, which vary throughout the process.

It may be shown, either from the functional equation in (3), or, as a consequence of known results in the theory of linear inequalities, that the maximization over the region depended by (4, 4) reduces to a maximization over the seven possible vertices marked in the figure above.

Of these seven, only the first five are actually possibilities, since vertices 6 and 7 effectively and the process-no new steel being produced.

The stagewise maximization probelm is thus quite trivial. The tabulation problem, however, remains non-trivial, because of the possibility of an expanding (c_1, c_2)-domain as R increases. The difficulty will be overcome in the following section.

REDUCTION IN DIMENSION AND MAGNITUDE

Let us show that we can reduce the problem to a sequence of no-dimensional probiems and simultaneously introduce shrinking transformations.

It is first of all clear from the linearity of all the constraints and production function that f_n (c_1, c_2) is a homogeneous function of c_1 and c_2 of the first degree.

Hence, for $c_1, c_2>0$, we have

$$\begin{aligned} f_n(c_1, c_2) &= c_1 f_n(1, c_2/c_1) \\ &= c_2 f_n(c_1/c_2, 1) \end{aligned} \tag{1}$$

It follows that we need comput only $f_n(1, x)$ or $f_n(x,1)$. Turning to (4.3), we have

$$
\begin{aligned}
f_R(1, c_2) &= \max_{\{s\}} [z_a + f_{R-1}(a_3 z_s, c_2 + a_4 z_m)] \\
&= \max_{\{s\}} \left[z_a + a_3 z_s f_{R-1}\left(1, \frac{c_2 + a_4 z_m}{a_3 z_s}\right)\right]
\end{aligned} \tag{2}
$$

We see, then, that caluculation of $f_R(c_1, c_2)$ for $c_2 \geq 0$, depends only upon a knowledge of $f_{R-1}(1, c_2)$ for $c_2 \geq 0$. This is the required reduction in dimensionality. However, we still face the difficulty of an expanding range for c_2.

In order to avoid this difficulty, let us show that we can compute $f_R(1, x)$ and $f_R(x, 1)$ for $0 \leq x \leq 1$, knowing $f_{R-1}(1, x)$ and $f_{R-1}(x, 1)$ for $0 \leq x \leq 1$.

Referring to (2), we have

$$
\begin{aligned}
f_R(1, c_2) &= \max_{\{z\}} \left[z_a + a_3 z_s f_{R-1}\left(1, \frac{c_2 + a_4 z_m}{a_3 z_s}\right)\right] \\
&\qquad \text{for } a_3 z_s \geq c_2 + a_4 z_m, \\
&= \max_{\{z\}} \left[z_a + (c_2 + a_4 z_m) f_{R-1}\left(\frac{a_3 z_s}{c_2 + a_4 z_m}, 1\right)\right] \\
&\qquad \text{for } a_3 z_s \leq c_2 + a_4 z_m
\end{aligned} \tag{3}
$$

Combining this equation with observation above concerning the maximization over vertices, we have a fairly simple computational scheme.

COMPUTATIONAL TECHNIQUE

The simplification introduced by the transformation technique of ch. 4 represent the most significant contribution of the particular study, with respect to the programming the results are these :

1) a reduction in time and space requirements from N^2 to $2N$ and
2) considerable further reduction afforded by the elimination of expanding grid.

Let us elaborate upon these two points.

Normally, the express all possible states of a system defined by two independent parameters (here c_1, steel stockpile, and c_2, mill capacity) it is necessary to construct a grid of $f_T(c_1, c_2)$,in (c_1, c_2) space, and then to interpolate over this 2-dimensional region to determine $f_T(c_1, c_2)$, the steel allocatable to auto production during an N-period process where the initial conditions are c_1' and c_2' and an optimal policy is pursued. This function is necessary for our recursive calculation of $f_{T+1}(c_1, c_2)$. Where the interval $[0, c_1]$ and $[0, c_2]$ are divided into N parts by this grid, we must compute and store N^2 values of $f_T(c_1, c_2)$ for future use.

The time requirement become ever more serious because of the extra logic needed when dealing with a 2-dimensional system. Hence the savings resulting from the reduction to one-dimensional form.

The possibility of an expanding grid is a serious obstacle in some dynamic programming processes. Non-mathematically the problem is this : To calculate the conditions at time t we must know in advance all possible state in which may find yourself at time $t+1$. In this particular application, to determine auto production over T periods, we must know auto production for $T-1$ periods for all allowable steel stockpiles and capacities. But after one period of production, either stockpile f_T (c_1, c_2) can be calculated in smaller than the region over which $f_{T-1}(c_1, c_2)$ is known. One must therefore begin an N-stage calculation by considering a large region in order to complete the calculation with a modest range of values for c_1 and c_2. The technique of 6. by passing the obstacle, represents a real and significant advance.

One further innovation in this problem, the optimization over a three dimensional region, bears mention. Techniques for the solution of general problems of the multi-dimensional nature have been little investigated. Here, of course, we are saved by the nature of the functions and have shown that only the vertices of the region need be considered. The coding technique used to determine and evalute the relevent vertices is diagrammed in figure 1.

The remainder of the problem is concerned with the calculation of a table of values of $f_T(c_1, c_2)$, the block transfer of this table, and its subsequent use in the derivation of table of $f_{T-1}(c_1, c_2)$.

Table 1 present an analysis of a typical set of results. The

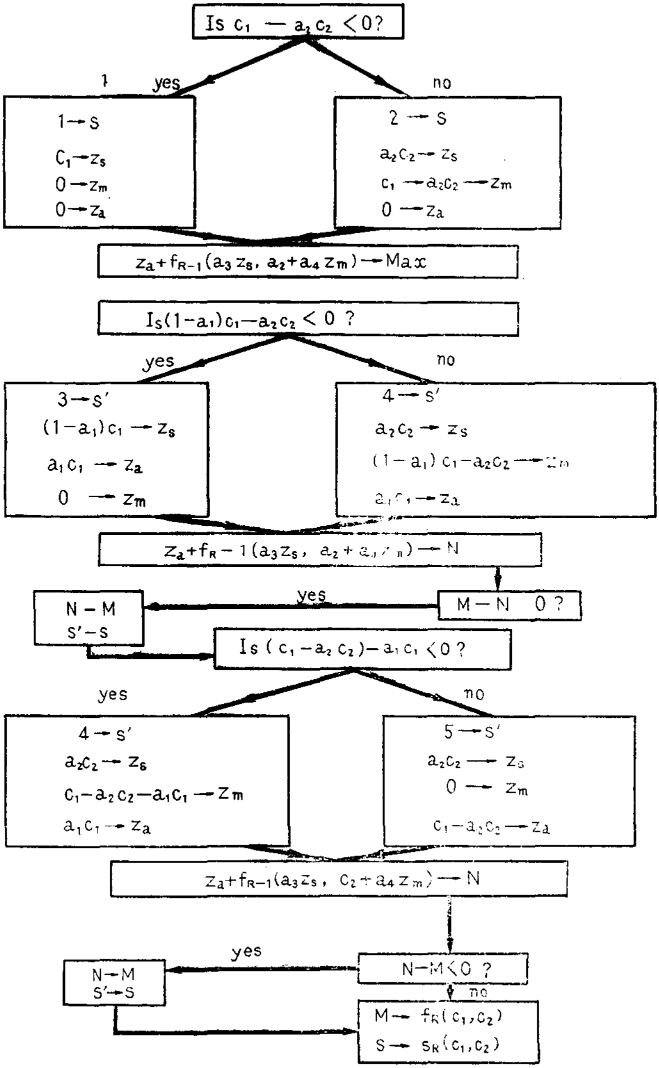
Is c1 — a2 c2 < 0?
yes
no
1 → s
C1 → zs
0 → zm
0 → za
2 → s
a2c2 → zs
c1 → a2c2 → zm
0 → za
za + fR-1(a3 zs, a2 + a4 zm) → Max
Is (1 − a1) c1 − a2 c2 < 0 ?
yes
no
3 → s′
(1 − a1) c1 → zs
a1 c1 → za
0 → zm
4 → s′
a2 c2 → zs
(1 − a1) c1 − a2c2 → zm
za + fR − 1 (a3 zs,
yes
N − M
s′ − s
M − N 0 ?
Is (c1 − a2 c2) − a1 c1 < 0 ?
yes
no
4 → s′
a2c2 → zs
c1 − a2 c2 − a1c1 → zm
a1 c1 → za
5 → s′
a2c2 → zs
0 → zm
za + fR−1(a3 zs, c2 + a4 zm) → N
yes
N → M
S′ → S
N − M < 0 ?
no
M → fR(c1, c2)
S → sR(c1, c2)

Table 1 15 stage process

$a_1 = .2$
$a_3 = 2$
$a_4 = .4$
$c_1 = 1$
$c_2 = 1$

	Information made available by calculation		Actual conditions using optimal policy		
Stage	Normalized condition at (c_1, c_2) beginning of stage	Vertex	Actual condition at beginning of stage		Steel allocated to auto production
1	(1, 1)	(1)	(1,	1)	0
2	(1, .5)	(2)	(2,	1)	0
3	(1, .7)	(2)	(2,	1.4)	0
4	(1, .5857)	(2)	(2.8,	1, 64)	0
5	(1, .642)	(2)	(3.28,	2.096)	0
6	(1, .611)	(2)	(4.192,	2.57)	0
7	(1, .626)	(2)	(5.14.	3, 22)	0
8	(1, .620)	(4)	(6.44,	3.988)	1.288
9	(1, .558)	(4)	(7.976,	4.4536)	1.595
10	(1, .586)	(4)	(8.9072,	5.2196)	1.781
11	(1, .573)	(4)	(10.4392,	5.9817)	2.008
12	(1, .579)	(4)	(11.9634,	6.9268)	2.393
13	(1, .576)	(4)	(13.8536,	7.9797)	2.771
14	(1, .577)	(4)	(15.9594,	9.2086)	3.192
15	(1, .577)	(4)	(11.4172,	10.6226)	3.683

Total allocation to auto production = 18.853

calculation generates the optimal choice of vertex at each stage for each initial condition and also lists the total steel allocation to auto-production achievable. To better display the results, a hand calculator was then performed, using the policy dictated by the computor, which shows the actual, unnormalised, growth of the system as a function of time. The sensitivity of the process was demonstrated by evaluating the return from a policy that was optimal in all but the first decision. An initial choice of vertex 3 results in an overall reduction in productivity of 8%.

DISCUSSION

Referring to Table 1 in the previous section, it will be observed that the optimal policy leads to a certain invariant state. As we shall show in a subsequent paper, this invariant state may be predicted in advance.

The existence of this state is related to a number of known results concerning the growth of an economic system over time, as we shall also discuss.

BIBLIOGRAPHY

1. Richard Bellman ; Bottleneck Problems, Functional Equations and Dynamic Programming, Econometrica
2. Richard Bellman ; Dynamic Programming of Continuous Processes, Rand Report R-271, 1954.

SUMMARY

In this paper, we consider the computational solution of a maximization problem arriving in the study of the utilization of interdependent industries. Under the assumption of proportional costs and returns, it is shown that the dimensionality of the problem can always be reduced by one, and all the transformations occuring can taken to be "shrinking transformations". This transformation overtly improves the efficiency of the method.