27. J. Kuczynski, *Die Geschichte der Lage der Arbeiter in den Vereinigten Staaten von Amerika seit 1898*, B. 1955.
28. Д. Кендрик, Тенденции производительности в США, М., Прогресс, 1967.
29. S. Kuznets, *Capital in the American Economy: Its Formation and Financing*, N.Y. 1961.
30. R. Goldsmith, *A Study of Savings in the United States, vol. III, Special Studies, Princeton*, 1956.
31. A. Grant, »Issues in Distribution Theory: The Measurement of Labor's Relative Share«, 1899—1929, *Rev. of Econ. and Stat.*, v. 45, N 3, 1963.
32. U.S. Department of Agriculture, *Agricultural Statistics*.
33. Р. Голдемит, Национальное богатство США в послевоенный период, М., Прогресс, 1968.
34. J. Gottsegen, »Revised Estimates of G.N.P. by Major Industries«, *Survey of Current Business*, 1967, N 4.

DYNAMIC PROGRAMMING AS A POTENTIAL PLANNING TOOL

1. The Scope of the Paper

This paper attempts to present some facts and examples relevant to the claim that dynamic programming is a potentially useful planning tool. I say »potentially« because first of all I am not entirely convinced that it is useful, although the more experience I gain the more convinced I become, and secondly dynamic programming has such a short history of application in economic planning that evidence on this point is not plentiful. True, such venerable researchers as Stone and his colleagues (1) made recourse to it briefly during the early stages of the Cambridge Growth Project, but this by itself hardly justifies the case for dynamic programming.

The important thing when investigating a new tool or technique is to obtain plenty of practical experience by using it. As Nemhauser has said, »It is absolutely necessary to solve problems to understand dynamic programming« (2). Over the past fifteen months, I have been endeavouring to achieve just this aim by applying dynamic programming to some of the problems which have occurred during research being carried out by the National Economic Planning Unit at Birmingham University. The main emphasis of this research, as several publications show (3), (4), (5), lies in the field of large optimization models, which are being solved using various decompositional techniques. As we shall see later, dynamic programming becomes rather inefficient when applied to large systems, generally losing out to more established techniques.

One cannot help getting the feeling when using dynamic programming, however, that such an inherently usable approach must have advantages for certain planning problems. The ease with which it can be used to obtain solutions to problems for which previously only approximations were possible quickly convinces the planner that there is something worth investigating here. Dangers loom large for the researcher who unwittingly strays too close to the boundaries of reasonability, or the user who expects too much too soon, and such hazards may be sufficient to persuade the impatient that no further effort is justified. This is not the opinion of this writer.

I hope to show, with the aid of three simple examples, that there are grounds for continuing to devote time and effort to the study of dynamic programming as a possible planning tool. This problem solving approach is supported by no less an authority than Bellman, the father and discoverer of dnamic programming, who said, »At the present time, we possess a number of powerful mathematical techniques for the analytic and computational solution of classes of problems in the field of mathematical economics. What is now needed is a systematic exploitation of these methods to provide a backlog of solved problems which will guide our subsequent research« (6).

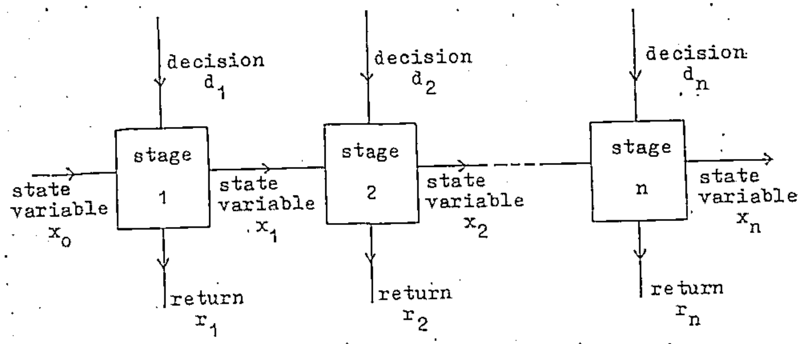
Since this statement, many applications have been made, but the majority seem to have been in fields other than economics. It is hoped that this paper will redress the balance slightly by looking at three problems from inventory, allocation and capital budgeting. The approach throughout is simplified where possible so as not to confuse the non-mathematical planner, although we are doing our best at Birmingham to eliminate this species. In the inventory problem especially, this means looking at a problem usually as stochastic-continuous in a deterministic-discrete way. By so doing, frequency functions and the like are avoided, and the essential features of dynamic programming are that much clearer.

The more discerning planner might also criticise the models in one or two cases. I would probably support him were that the purpose of this paper, but it is not. The approach is to be considered above all else, and discussions of the models must await another opportunity. When speaking a foreign language, it is reassuring to know at least something about the topic of conversation, so that the learner can concentrate on the grammar; the models, are, therefore, familiar and uncomplicated.

One more point which must be made is that I have some qualms about my passport being valid for the country in which I find myself. Economica contains far too many regions where the dialect is incomprehensible to a simple mathematician. I trust that the reader will forgive the odd crude phrase here and there, and that my strange turn of tongue allows the essentials nevertheless to come through.

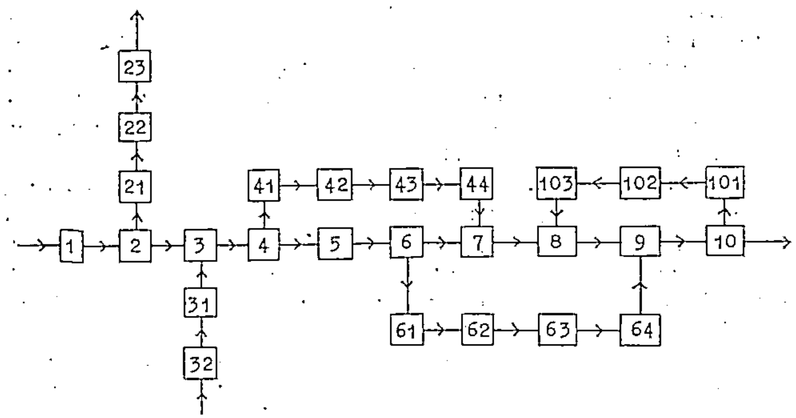
Before beginning the treatment of the three examples, a few introductory remarks on dynamic programming in general are probably in order. Firstly, the term itself is often misleading to the casual reader. Perhaps recursive optimization or sequential programming would better describe what is actually going on. The problem being considered need contain no reference to time, as »dynamic« might imply, although many of the planning problems suitable for treatment are naturally of this type.

To take the novice further, the next words he must learn are stage, state, return, and decision. The following diagram contains all four, and speaks for itself:



This n-stage decision problem, where the decisions and returns have been decomposed so that at any stage there is only one decision to be made, based on the information available in terms of the state variable, is the core problem of dynamic programming. Any problem to which dynamic programming can be applied has this basic structure, and it is essential therefore to appreciate its constituent parts, and how they inter-relate.

The three examples discussed in this paper all have this structure as their foundation, but it would be wrong to convey the impression that this is typical. Complex structures can be built up, like the one shown in the next diagram:



This problem has divergent branches, convergent branches, feedback, and feedforward, but each of the branches considered separately displays the same basic structure. By treating these branches in the appropriate order, the optimum may be obtained with no major modification of the approach whatsoever.

This then defines the scope of my paper; we shall be looking at deterministic, discrete, optimization problems in two of the exaples, and at a deterministic, continuous optimization problem in the other. I make no apologies for omitting to recite the well-known optimality principle found in all texts on dynamic programming; I regard the principle as something any self-respecting planner would consider obvious and unnecessary to the understanding of the subject.

Finally, I would acknowledge the benefits received from selected reading of the works of Bellman (7), (8), Nemhauser (9), and Larson (10). For me, they are all that the novice requires to make his journey into the realm of dynamic programming a rich and pleasant experience.

2. *An Inventory Problem*

The contribution, in value terms, of the petroleum sector to the total exports of Venezuela between 1952 and 1962 varied between 86 per cent and 95 per cent (see (11)). With this as background, let us imagine a situation in which the Venezuelan planning office seeks to coordinate the production of petroleum over a four month period with estimated demand and available storage facilities at the export points.

To keep the exposition simple, consider only four production possibilities, and five storage possibilities, each of which has a different associated cost, as shown below: (the data is fictitious)

monthly production possibilities (m. gallons)	0	1	2	3	
associated costs (m. bolivares)	200	250	350	600	
monthly storage possibilities (m. gallons)	0	1	2	3	4
associated costs (m. bolivares per month)	25	30	40	55	80

Estimated exports we will assume to be 1, 5, 2 and 2 milion gallons in the four months under consideration. We will also assume that two million gallons are initially in storage, and that one million gallons are required to be in storage at the end of the four month period. Storage costs are applied to stocks on hand at the beginning of the month (other conventions could be used here without unduly affecting the approach). Exports are assumed to take place on the last day of every month.

The model for this problem may be written

$$\min z = \sum_{j=1}^{4} [C_j(x_j) + G_j(i_{j-1})]$$

$$\text{subject to } i_{j-1} + x_j - i_j = d_j, \quad j = 1, 2, 3, 4$$

$$x_j \leqslant 3, \quad j = 1, 2, 3, 4$$

$$i_j \leqslant 4, \quad j = 1, 2, 3$$
$$i_0 = 2, \quad i_4 = 1$$
$$x_j, \ i_j \text{ integers, all } j.$$

where

i_{j-1} = inventory at beginning of month j;

i_j = inventory at beginning of month $j + 1$ (equal to stocks at end of month j after exports have taken place);

x_j = production in month j;

d_j = demand in month j;

$G_j(x_j)$ = month j production cost if x_j is produced;

$G_j(i_{j-1}$ = month j inventory cost if $i\ j-1$ is initially in stock.

This model allows for different production and inventory costs in every month, but we have assumed

$$C_j(x_j) = C(x_j) \qquad G_j(i_{j-j}) = G(i_{j-1})$$

so as not to be over-burdened with data. This assumption does not affect the viability of dynamic programming as a solution technique.

Inserting the known demand and inventory figures, the model is

$$\min z = 40 + C(x_1) + G(i_1) + C(x_2) + G(i_2) + C(x_3) + G(i_3) + C(x_4)$$

$$\begin{aligned} \text{subject to } \ x_1 - i_1 &= -1 \\ i_1 + x_2 - i_2 &= 5 \\ i_2 + x_3 - i_3 &= 2 \\ i_3 + x_4 &= 3 \end{aligned}$$

$$0 \leqslant x_1 \leqslant 3, \ 0 \leqslant i_1 \leqslant 4, \ 0 \leqslant x_2 \leqslant 3, \ 0 \leqslant i_2 \leqslant 4, \ 0 \leqslant x_3 \leqslant 3, \ 0 \leqslant i_3 \leqslant 4,$$
$$0 \leqslant x_4 \leqslant 3$$

$$\text{all } x_j, \ i_j \text{ integers}$$

Any integer programming algorithm which first obtains a continuous optimum would be no use here, since $C(x_j)$ and $G(i_j)$ are not defined for x_j, i_j continuous. The method of Lagrange is also ruled out because of the integer restriction. Total enumeration is a possibility and since the ranges of x_j and i_j are small, one may be tempted to pursue this idea. There are seven decision variables, four of which have four feasible values, and three of which have five feasible values.

$\therefore$ total number of combinations = $(4^4) \cdot (5^3)$ = 32000 Many of these are infeasible, and by intelligent use of the constraints, this number can be considerably reduced; from the last constraint, for example,

$$i_3 \leqslant 3$$

and summing all four constraints gives

$$x_1 + x_2 + x_3 + x_4 = 9$$

But there are still many combinations remaining to be checked for feasibility, followed by objective function evaluations and comparisons. The dynamic programming approach is to decompose the problem into four subproblems, in each of which we consider combinations of the current month's inventory level and the previous month's level (production is then automatically determined, since demand is given).

Four tables are obtained:

		i_1: 0	1	2	3	4
i_0	2	—	240	290	300	640
		—	240	290	390	640

(— indicates an infeasible combination of state and decision variable)

Month one

		i_2: 0	1	2
i_1	1	—	—	—
	2	930	—	—
	3	795	1045	—
	4	970	1070	1320
		795	1045	1320

Month two

		i_3: 0	1	2	3
i_2	0	1170	1420	—	—
	1	1325	1425	1675	—
	2	1560	1610	1710	1960
		1170	1420	1675	1960

Month three

		i_4
		1
	0	1795
	1	1800
i_3	2	1965
	3	2215
		1795

Month four

In this problem each stage is one month, the state variables are previous stage inventory levels, the decision variables are current stage inventory levels, and the stage returns (shown in the bodies of the tables) are cumulative costs. To illustrate how a stage return is obtained, consider $r(i_2, i_3) = r(1,1)$ in month three. The cost figure derives from

$G(i_2) = G(1) = 30$

$+ \; C(x_3) = C(2 + i_3 - i_2) = C(2) = 350$

$+$ minimum cost of starting in month

three with $i_2 = 1$ (from second table) $= 1045$

TOTAL $= 1425$

The optimum solution is

$$i_3 = 0$$

from which it can be discovered that

$$i_2 = 0$$

leading to

$$i_1 = 3$$

By substitution,

$x_1 = 2,\ x_2 = 2,\ x_3 = 2,\ x_4 = 3$, cost $= 1795$

Further benefits to be derived from this approach include sensitivity analysis. By deepening the month one table, we can cost all possible values of i_0. Tables two, three and four remain the same size, and the majority of figures in them are unaffected. Similarly, table four may be widened to calculate the effect on the solution of different i_4 values. Tables one, two and three are exactly the same in this case.

We have also obtained a large number of sub-optimal solutions. The second best solution is

$i_4 = 1,\ \ i_3 = 1,\ \ i_2 = 0,\ \ i_1 =$ cost $= 1800$

If secondary objectives exist in this problem, then this solution may be preferred, since the increase in cost over the true optimum is almost negligible.

3. An Allocation Problem

The second example is a regional allocation problem, constructed on the lines laid down by Johansen (12). Again a fictitious planning framework has been constructed to lend some meaning to the problem. This model is in deterministic, continuous variables, and is of the type usually solved by application of the decomposition principle (13).

Suppose that the government of an African country is faced with the problem of allocating 15 units of capital between its northern and eastern copper producing regions. The eastern region, which also produces iron ore, requires one unit of capital per unit of copper output, and three units of capital per unit of iron ore produced. This region has available seven units of labour, and nine units of mining equipment. Per unit of output, use of these »resources« is three and two units of labour, and two and one units of equipment, for copper and iron, respectively.

The northern region requires two units of capital per unit of copper output, and three units of capital per unit of zinc output. Labour and equipment are the basic inputs to these two mining processes, and usage is one and five units of labour, and two and three units of equipment, per unit of copper and zinc mined, respectively. The region has eight units of labour and five units of equipment.

The eastern region requires at least two units of copper output for full utilization of available smelting plant, and the northern region requires at leats one unit of copper output for similar reasons. Transportation of copper from the north to the east is not allowed, although version of the problem in which transportation takes place can be dealt with.

From this information, a linear programming model can be constructed:

max $R = 2x_1 + 3x_2 + 2y_1 + y_2$ (revenue)

subject to $x_1 + 3x_2 + 2y_1 + 3y_2 \leqslant 15$ (capital)

eastern region:
$3x_1 + 2x_2 \leqslant 7$ (labour)
$2x_1 + x_2 \leqslant 9$ (equipment)
$x_1 \geqslant 2$ (min. copper output)

northern region:
$y_1 + 5y_2 \leqslant 8$ (labour)
$2y_1 + 3y_2 \leqslant 5$ (equipment)
$y_1 \geqslant 1$ (min. copper output)

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ y_1 \geqslant 0,\ y_2 \geqslant 0$$

(copper) (iron) (copper) (zinc)

where

x_1 = eastern region copper output;

x_2 = eastern region iron ore output;

y_1 = northern region copper output;

y_2 = northern region zinc output;

R = total regional revenue.

Application of Dantzig-Wolfe decomposition to this problem would result in a master problem having three constraints, generating objective functions for two three-constraint subproblems. These three problems may need solving a number of times before optimality is achieved.

The dynamic programming approach is exceedingly simple: from the full problem we obtain a two-stage decision problem, each stage consisting of two decisions. These two subproblems will be solved only once in order to achieve optimality. The stage one problem is

$$\max R_1 = 2x_1 + 3x_2 \quad \text{(eastern region revenue)}$$

$$\text{subject to } x_1 + 3x_2 \leqslant y \quad \text{(capital)}$$

$$3x_1 + 2x_2 \leqslant 7 \quad \text{(labour)}$$

$$2x_1 + x_2 \leqslant 9 \quad \text{(equipment)}$$

$$x_1 \geqslant 2 \quad \text{(min. copper output)}$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0$$

(copper) (iron)

where

y = total amount of capital allocated to the eastern region.

The variable y is the state variable referred to in an earlier section of this paper. During stage one y is not known, and so we allow it to vary between zero and fifteen.

The stage one problem is a parametric linear programming problem, y here being the parameter. The solution is not very difficult to obtain:

$$\text{for } 2 \leqslant y \leqslant \frac{7}{3},\ x_1 = y,\ x_2 = 0,\ R_1 = 2y$$

$$\text{for } \frac{7}{3} \leqslant y \leqslant \frac{3}{2},\ x_1 = \frac{21 - 2y}{7},\ x_2 = \frac{3y - 7}{7},\ R_1 = \frac{21 - 5y}{7}$$

$$\text{for } \frac{3}{2} \leqslant y \leqslant 15,\quad x_1 = 2,\quad x_2 = {}^1/_2,\ R_1 = \frac{11}{2}$$

The stage two problem is then formed:

$$\max R = 2y_1 + y_2 + R_1(y) \quad \text{(total revenue)}$$

$$\text{subject to } 2y_1 + 3y_2 + y = 15 \quad \text{(capital)}$$

$$y_1 + 5y_1 \leqslant 8 \quad \text{(labour)}$$

$$2y_1 + 3y_2 \leqslant 5 \quad \text{(equipment)}$$

$$y_1 \geqslant 1 \quad \text{(min. copper output)}$$

$$y_1 \geqslant 0,\quad y_2 \geqslant 5,\quad 2 \leqslant y \leqslant 15$$

(copper) (zinc) (capital)

The interpretation of this problem is not difficult, and neither is the solution. The objective function is piecewise linear, and several algorithms exist for solving such problems (14, 15). Using the Dantzig version, the solution obtained was

$$y_1 = \frac{5}{2},\ y_2 = 0,\quad y = 10,\quad R = \frac{21}{2}$$

from which the overall solution is easily derived:

$$x_1 = 2,\quad x_2 = {}^1/_2,\quad y_1 = \frac{5}{2},\ y_2 = 0,\quad R = \frac{21}{2}$$

The optimism one may feel after solving the problem in such a straightforward fashion is quickly dissipated at the realisation that the number of joining constraints is rather critical. Two and three constraints could possibly be handled without undue difficulty, but more than three causes strains in the parametric algorithm. Various devices have been proposed to overcome this shortcoming, among them being the use of a Lagrange multiplier (16), or multipliers, to transfer some of the joining constraints to the objective. Apart from the fact that choosing the multipliers is by no means a simple matter, this manoeuvre only reduces the dimensionality problem by two or three more constraints.

This excursion into the major league of linear programming is not successful from the dynamic programmer's viewpoint. The impression remains, however, of a rather beautiful marriage (albeit a short one!) between a structured linear programme and the sophisticated dynamic programming approach. The trouble lies with the parametric algorithm, not with dynamic programming.

4. A Project Planning Problem

For our third problem we will consider the allocation of certain amounts of capital over time to a number of projects. The model and the approach to be described are due to Nemhauser and Ullmann (17).

A suitable general form of the model with which to begin is

$$\max \sum_{k=1}^{N} \sum_{l=1}^{Lk} \left[\sum_{t=0}^{T} \frac{s_{tkl} - y_{t+1,kl}}{(1+r)^t} \right] d_{kl}$$

$$\text{subject to } \sum_{k=1}^{N} \sum_{l=1}^{Lk} A_{kl}\, d_{kl} \leqslant C$$

$$\sum_{l=1}^{Lk} dkl = 1, \ k\ 1,2, \ldots, N$$

$$d_{kl} = 0 \text{ or } 1, \quad 1 = 1, 2, \ldots, L_k, \quad k = 1, 2, \ldots, N$$

where N = the number of investment projects, called $P_1, P_2, \ldots, P_N$;

T = the planning horizon, or length of the longest project;

s_{tkl} = the return, realised at the end of period t, from project P_k operated at level 1;

y_{tkl} = the investment required at the beginning of period t for project P_k to be operated at level 1;

r = a discount factor, usually the cost of capital;

$C = (c_1, c_2, \ldots, c_T)$ = a vector of capital availabilities

$A_{kl} = (a_{1kl}, a_{2kl}, \ldots, a_{Tkl})$ = a vector of capital outlays required to run project P_k at level 1;

d_{kl} = 1 if project P_k is run at level 1, and 0 otherwise.

We are dealing with an allocation model similar to that described in the last section, but with the important differences that all the decision variables are of the zero-one type, and the objective function is highly irregular. A numerical example of this particular model, solved by dynamic programming in its simplest form, would be impossible to handle for anything but the smallest problems. Tables of the type constructed in the Venezuelan oil example would assume massive proportions. To demonstrate the magnitude of this difficulty, a ten project, five period example in which $c_t = 25$ for all t, and $L_k = 5$, all k, would require ten tables each of size 5 × 265

A moment's thought on the oil example, however, is sufficient to see that usually only one particular value in each row of the tables is important, and that is

$$f(x_j) = \min_{d_j} \left\{ q(x_j, d_j) \right\}$$

where x_j is the state variable, d_j is the decision variable, and $q(x_j, d_j)$ is the cumulative return from this combination of the two variables. Storing $f(x_j)$ and $d_j(x_j)$ (the decision responsible for the particular value of $f_j(x_j)$) is, therefore, sufficient. In the project example this is of apparently little help, since $f(x_j)$ is a vector with 26^5 elements.

However, the functions $f(x_j)$ have an extremely useful property in this case which leads to a vast reduction in storage requirements. Let us invent some data to illustrate this. Consider a one period three project example, each project being of the zero-one variety (i.e. $L_k = 2$, all k). The model may be reduced to

$$\max \sum_{k=1}^{3} b_k\, d_k$$

$$\text{subject to } \sum_{k=1}^{3} a_k\, d_k \leqslant c$$

$$d_k = 0 \text{ or } 1, \quad k = 1, 2, 3$$

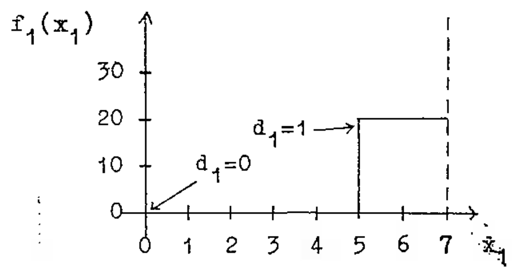
and assume that

$B = (b_1, b_2, b_3) = (20, 10, 40)$

$A = (a_1, a_2, a_3) = (5, 3, 2)$

$c = 7$

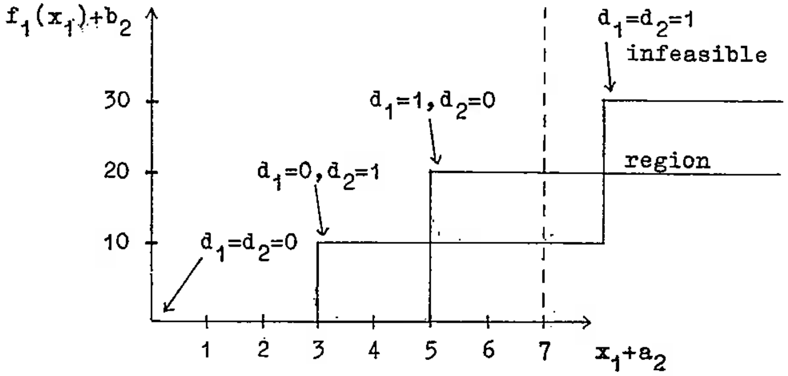
This is a three-stage decision problem in which at stage k decision d_k is taken, $k = 1, 2, 3$. The state variable at stage k is x_k, this being the amount of capital still remaining. Thus, at stage one there might be any value of x_k between 0 and 7. A graph of optimal return against total availability of capital (i.e. a graph of $f_1(x_1)$ against x_1) is easily drawn:



GRAPH ONE: Stage one

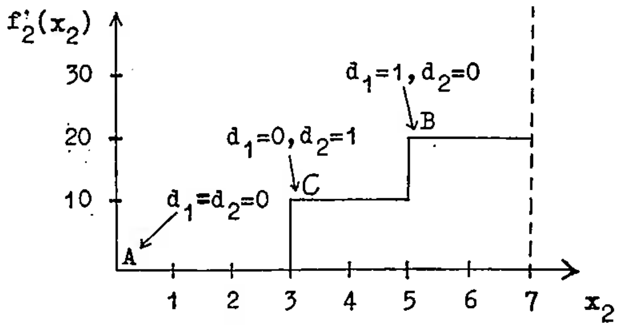
The return is zero if less than 5 units of capital are available, and is twenty if 5, 6 or 7 units are available. The graph is shown as continuous when it is clear that only integer values are feasible, but this is to aid the understanding of a concept to be introduced in a moment. Only two points are important on this graph: $(f_1(x_1), x_1) = (0,0)$ and $(20,5)$.

Moving on to stage two, we have to decide whether to run project two or not, and we must bear in mind at the same time that $f_1(x_1)$ is to be included in the calculations. Clearly $f_1(x_1) + b_2$ is the return and $x_1 + a_2$ the outlay required to run project two in addition to the decision taken at stage one. That is, we add (b_2, a_2) to the two step points of graph one:



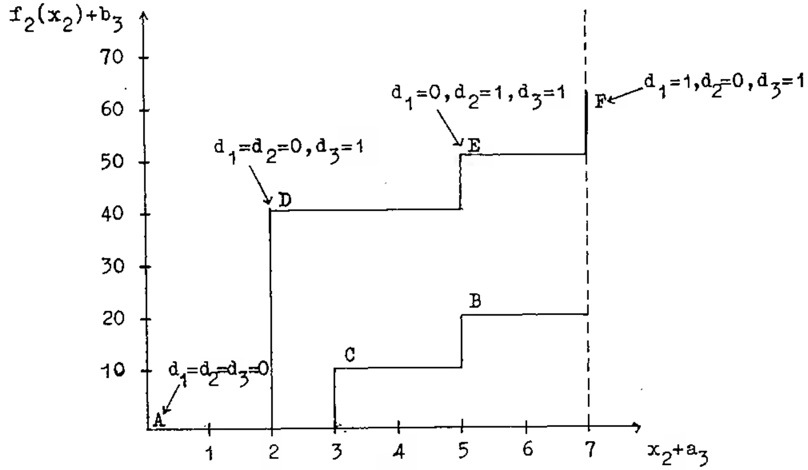
GRAPH TWO: Stage one plus $d_2 = 1$.

The point at which we would most like to be on this graph is reached by starting at (0,0), moving right until we come to a step, climbing the step, moving right, and so on. We stop moving right if the next barrier is the boundary of the feasible region. We then move back to the top of the last step, and pronounce this to be the best point. The graph can be tidied up to look like



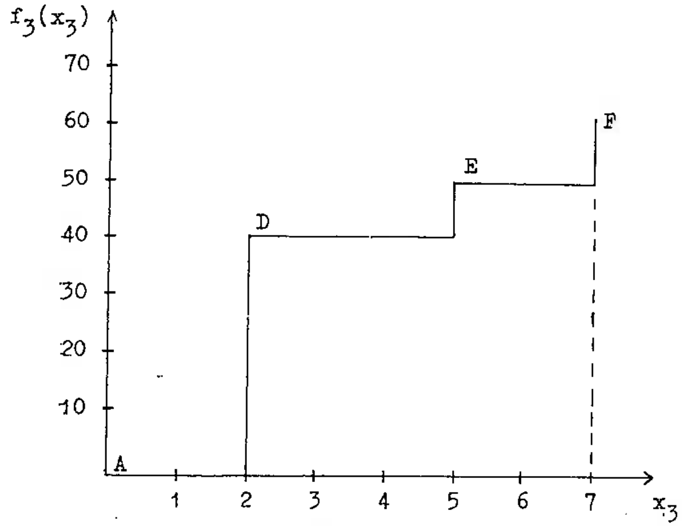
GRAPH THREE: Stage two

Proceeding in the same manner, we add $(b_3, a_3) = (40,2)$ to $(f_2(x_2), x_2) =$ $= (0,0)$, $(10,3)$, $(20,5)$:



GRAPH FOUR: Stage two plus $d_3 = 1$

and the optimum solution is point F, at which $d_1 = 1$, $d_2 = 0$, $d_3 = 1$, and which represents an outlay of seven units of capital and a total net return of sixty.



GRAPH FIVE: Stage three

5. Concluding Remarks

The objective of this paper is to »popularise« dynamic programming in an economic planning context. In seeking to achieve this objective, one runs the risk of under-selling the technique by making the exposition too simple and unrealistic. On balance, this risk appears justified when the opposite extreme is considered, that of building a firm mathematical foundation before applications are attempted. Bellman's papers are strongly mathematical in places, and the concentration is almost entirely on achieving recursion equations which, in many cases, cannot be solved.

One could, therefore, usefully spend one's time in trying to discover ways of overcoming difficult recursion equations, making use of ever more mathematics. Eventually the equations would yield to some all-powerful new approach, but what of the applications side in the meantime?

This author is convinced that planning situations must be sought in which the recursion equations may be simplified. Stocks of solved problems should then be built up, and, from such evidence, ways of extending the planning problems suggested. Hopefully, the recursion equations would also be extended, retaining their desirable properties from a planner's point of view. Those of us working in this direction should meet up with the mathematicians at some point in time. It remains to be seen whether or not a satisfactory approach rate can be maintained.

University of Birmingham, U.K. *David LIGGINS*

REFERENCES

[1] L.J. Slater, »A Dynamic Programming Process«, *The Computer Journal*, Vol. 7, No. 1, April 1964.

[2] G.L. Nemhauser, *Introduction to Dynamic Programming*, J. Wiley, New York, 1966, p. viii.

[3] D. Liggins, *The Foreign Trade Subproblems of an International Cooperation Model*, Jahrbuch der Wirtschaft Osteuropas, 1970 (forthcoming).

[4] T.O.M. Kronsjö, »International and Interregional Economic Cooperation and Planning by Linked Computers«, in: Allen J. Scott (Ed.), *Studies in Regional Science, London Papers on Regional Science*, Pion Ltd., 1969, pp. 181—200.

[5] A.C. McKay, *Experience Gained Through the Application of the Double Decomposition Technique to Large Linear Programming Problems* (forthcoming).

[6] R. Bellman, *Dynamic Programming and Mathematical Economics*, RAND Memorandum RD-3539-PR, March 1963, p. 8.

[7] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1957.

[8] R. Bellman and S.E. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, Princeton, New Jersey, 1962.

[9] G.L. Nemhauser, *Introduction to Dynamic Programming*, J. Wiley, New York, 1966.

[10] R.E. Larson, *State Increment Dynamic Programming*, American Elsevier Publishing Co., New York, 1968.

[11] F.D. Levy, *Economic Planning in Venezuela*, Praeger, New York, 1968.

[12] L. Johansen, »Regionalokonomiske problemer belyst ved linear programmeringsteori«, *Sosialokonomen*, February 1965.

[13] G.B. Dantzig and P. Wolfe, »The Decomposition Algorithm for Linear Programming«, *Econometrica*, Vol. 29, No. 4, October 1961.

[14] G. Hadley, *Nonlinear and Dynamic Programming*, Addison-Wesley, Reading, Mass., 1964, pp. 104—109.

[15] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, 1963, pp. 482—486.

[16] H. Everett, »Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resource«, *Operations Research*, 11, pp. 399—417.

[17] G. Nemhauser and Z. Ullmann, »Discrete Dynamic Programming and Capital Allocation«, *Management Science*, Vol. 15, No. 9, May 1969, pp. 494—505.

[18] H.M. Weingartner, »Capital Budgeting of Inter-related Projects: Survey and Synthesis«, *Managemet Science*, Vol. 12, No. 7, March 1966, pp. 485—516.

[19] Z. Ullmann, *Dynamic Programming and the Capital Allocation Problem*, Doctoral Dissertation, The Johns Hopkins University, Baltimore, 1967.

THE STRUCTURE OF WAGES IN YUGOSLAVIA*

Decisions about the wages paid to labor are among the more difficult and delicate tasks faced by a worker managed enterprise. There has been substantial theoretical speculation about the nature of this decision and its impact on the economy but virtually no empirical examination of the issue. The purpose of this paper is to analyze the problem of wage determination, with particular attention devoted to the *wage differentials* that derive from the wage determination process under the Yugoslav system of workes' management.

Interskill Wage Structure. The Yugoslav worker receives his wage in two parts — a *fixed wage* which he periodically receives as compensation for his labor input and a *variable wage* which he receives at the end of an accounting period. The latter represents a supplementary wage payment out of the current surplus of the enterprise.[1])

* This paper is part of a larger study of workers' management and wage differentials in Yugoslavia, financed by the Comparative Economics Program of the Universty of Michigan. A substantial portion of the research was conducted in Yugoslavia during the 1967—68 academic year at the Institute of Economic Studies. I appreciate especially the valuable comments of Branko Horvat, Morris Bornstein, Alexander Eckstein, Harold M. Levinson, and Dallas L. Jones. None of these individuals or institutions bears any responsibility for the content of this paper.

[1]) The way in which the fixed and variable wages are established in the enterprise is discussed in Howard M. Wachtel, »Workers' Management and Interindustry Wage Differentials in Yugoslavia«, (mimeographed, 1970). Some general questions concerning the nature of the wage decision are discussed in Howard M. Wachtel, »Wages in a Labor Managed Economy: The Yugoslav Case«, *Florida State University Slavic Papers* (forthcoming).