BUILDING MATHEMATICAL MODELS IN DYNAMIC PROGRAMMING

LIANA RODICA PATER
UNIVERSITATEA "TIBISCUS" TIMISOARA, lianapater@yahoo.com

Abstract:
In short, we can say that dynamic programming is a method of optimization of systems, using their mathematical representation in phases or sequences or as we say, periods. Such systems are common in economic studies at the implementation of programs on the most advanced techniques, such as for example that involving cosmic navigation. Another concept that is involved in the study of dynamic programs is the economic horizon (number of periods or phases that a dynamic program needs). This concept often leads to the examination of the convergence of certain variables on infinite horizon. In many cases from the real economy by introducing updating, dynamic programs can be made convergent.

Key words: *dynamic, programming, optimal, decision, stage*

JEL classification: C61

Intoduction

Management problems, in most cases, have individual hypotheses as starting points involving a single decision, which possibly can be further optimized, resulting in a generally applicable model. However, this requires the fragmentation of more complex problems, reducing it to simple cases, elementary, each of these cases being included in a model whose optimal solution is known. Finally, by assembling the partial results thus obtained one can build the final solution of the problem. Disadvantages of this method of solving the problem management are:

1. great time spent solving the problem;
2. determination of non-optimized solutions by the application of models of performance for each elementary event, while the assembly results may not be as rigorously so that the final solution to the optimal ([13]).

In management are considered, however a number of decisions, each one of these decisions affecting future decisions. The tool used to solve certain types of sequential decision problems is called *dynamic programming*.

There is no model for solving dynamic programming problems, so these problems are classified into groups, each group having its own form and solution method. However, formulating hypotheses and reasoning required to solve a dynamic programming problems are about the same (see [3, 6, 7, and 8]).

One of the most effective tools for optimum search function attached to the economic phenomenon is relatively new and has been highlighted by American mathematician *Richard Bellman*. This tool is known today as dynamic programming. In short, we can say that dynamic programming is a method of optimization of systems, using their mathematical representation in *phases* or *sequences* or as we say, *periods*. Such systems are common in economic studies at the implementation of programs on the most advanced techniques, such as for example those involving cosmic navigation ([1, 2, 4, 5, 11]). The starting point of this method is the "theorem of optimality" and most general form can be stated as a principle, "the principle of optimality".

When it comes to regulating production and storage of materials, equipment management of mineral exploration, investment, not to mention the macroeconomic issues such as national planning, the sequential nature of the problem justifies the use of appropriate mechanisms to allow optimization calculations to clarify issues and to introduce specific concepts such as decision criterion, policy or strategy to follow, the influence of information on the quality of decisions.

Another concept that is involved in the study of dynamic programs is the *economic horizon* (number of periods or phases that a dynamic program needs). This concept often leads to the examination of the convergence of certain variables on infinite horizon. In many cases from the real economy by introducing *updating*, dynamic programs can be made convergent.

1. **The nature of dynamic programming**

Overall, the initial problem is divided into smaller problems. Segmentation of complex issues in simple problems results from a series of decisions. Each of the problems created are called simple stage problems. In a matter of multi-stage decision, a series of interrelated decisions already exist or will be constructed. For example, the decision about how much it will cost us preventive maintenance car this year is interrelated with the decision to keep it or sell it next year. If our intention is to sell the car next year it will require less maintenance this year, if we intend to keep it for several years there must be allocation of money for inspections and repairs (see [7, 8, 11, 12]).

A tactical dynamic programming refers to a predetermined plan, a complete course of action selection in all possible circumstances. Optimal dynamic scheduling indicates a tactic to solve the problem, which is the best of all possible tactics for the entire problem from all tactics used for sub-problems. The basic idea - the principle of optimality in dynamic programming analysis is based on Bellman's principle of optimality:

An optimal policy has the property that for any initial state and any initial decision, the remaining decisions must constitute an optimal policy on the state resulting from the initial decision.

The importance of the principle of optimality is that starting from a given stage, optimal tactics (decision) for the remaining stages of the system state depends only on current state and does not mean that the system was used to reach that state (so, best tactic is independent of the tactics adopted at previous stages).

2. **Prototype dynamic programming problems**

Unlike most mathematical models, there are no standard recursive relations in dynamic programming. It is therefore impossible to use a general computational method (e.g. *Simplex method in linear programming*). However it is possible to classify the dynamic programming problem "families" (or prototypes) and build one specific computational procedure for each. These prototypes are:

1. Allocation processes. These processes are segmented into basic problems of allocation.

2. Multi-periodical processes. These processes are segmented from the start, with two or more periods of time. These are known as *planning processes*.

3. Process-network. Networks can often be viewed as dynamic programming problem and solved as such.

4. Multi-stage production processes. These problems arise in industrial production planning.

5. Feedback control processes. Feedback type problems can be encountered in electronics, aviation, automation, etc.

3. **An example of investment**

The allocation of resources is a major problem for all organizations. When dealing with a linear objective function, the problem can be presented as a linear programming problem. However, in many cases, the formulation of mathematical programming models lead to nonlinear problems with procedural solutions which are either difficult or expensive. Dynamic programming provides an improved technique to solve such complicated cases.

The example that follows illustrates how managerial decision process uses dynamic programming to make a calculation of investment needed to build three power plants by the company "Termoelectrica". Managerial staff of the unit wants to allocate $ 4 million in the three plants. It has been already decided that the investment in each center can be between 0 and 4 million dollars (investment is made in units of one million dollars).

Each power plant has submitted the annual revenue forecast corresponding to the various levels of money invested. This forecast is given in Table 1. For example, an initial investment of $ 2 million in plant A will generate an annual income of $ 5 million. The problem is to determine the optimal allocation of money for each center individually, in order to maximize the overall projected annual income. This problem can be solved by linear programming method, primarily because it is a programming problem and second, incomes are a nonlinear variable.

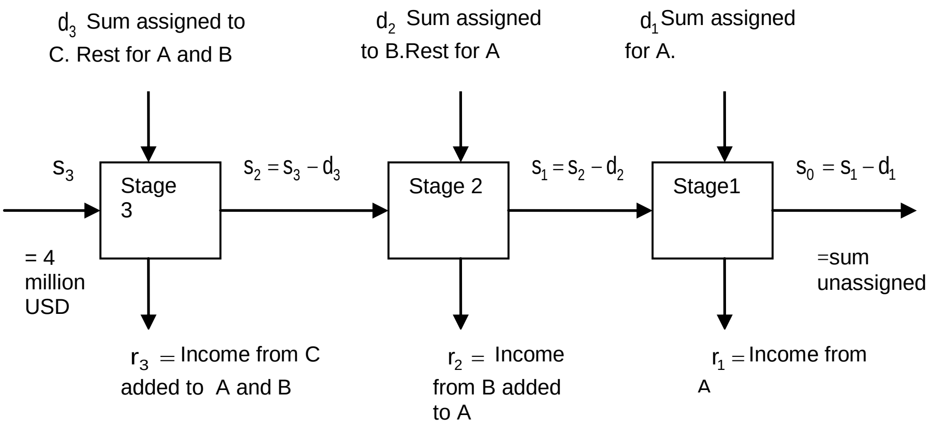
Figure 1: Alternative investment S.C. Termoelectrica S.A.

Amount (in million)	Annual income (reward) (in million dolars) coressponding to:		
	Plant A	Plant B	Plant C
0	0	0	0
1	0,2	0,3	0,4
2	0,5	0,6	0,9
3	1,5	1,2	1,1
4	1,4	1,5	1,6

Hypothesis: The problem is divided into three stages; each stage represents one allocation to a Plant. This means that the process will be viewed as a series of sub-decisions. Relations between stages are represented in Figure 2. We apply the inverse approach by an allocation process first to Plant A (arbitrarily considered as "last" Plant), then to Plant B and then to Plant C.

States: In each stage, there are five possible states, assigning 0, 1, 2, 3 or 4 units.

Figure 2. Resourse allocation problem



Solution

Stage I : In this stage, for Plant A will be available from 0 to 4 million dolars (sum denoted by s_1). The income calculated from investment in Plant A are given in Figure 3. The optimal policy is: if there are available 0,1,2, or 3 units, the best solution is to allocate all income. If there are available 4 units then best would be to allocate just 3 million dolars because optimal income obtained from 3 million is grater than the income obtained of 4 million (an unusal situation but nevertheless possible). This information can be found in Figure 3., where the last column is composed of the greatest (optimal) reward corresponding to each line. Underlined numbers are the greatest and they represent the optimal policy (optimal decision) (see [10]).

Stage II : In this stage is necessary to determine the ways to divide the available sums between A and B. In the sequel, we'll denote the available amount for allocation to A and B by s_2.

Thus, the amount s_2, Plant B obtain d_2 while the amount s_1=s_2-d_2 is available to allocate to Plant A in the best way possible, as already has been calculated in stage I.

For each value of s_2 (0, 1, 2, 3 or 4) there are several allocation alternatives, which must all be considered. This can be done by examining all five possible states.

For state s_2=0: not allocate anything, there is no income.

For state s_2=1: you can assign both 1 for B and 0 for A (total income being 3+0=3) or 0 for B and 1 for A (total income being 0+2=2). Clearly, 1 for B es is the best allocation, i.e., if 1 million remains to be allocated between A and B, it should be received by Plant B. This can be seen easily from Figure 3. Figure 3 includes all the calculations for the remaining states. It also shows calculations in each state and the total optimal reward. In general, calculations from the contents table refer to *total reward*, which is the sum of immediate reward and optimal reward from stage I.

Example for state s_2=3: Line 3 of Figure 3. is calculated as: once all total rewards are calculated for each line, the greatest of these is selected and designated as total optimal reward.

Stage III: In this stage, an allocation decision is made for Plant C and the remainder is allocated *in the best possible way* between A and B, according to the policy described in stage II. In this final stage just a state will be presented.

Example for state s_3=4*:* calculations are presented in Figure 3 in the standard manner used earlier. Thus, the best allocation is: d_3=1, d_2=0, d_1=3; i.e. C is assigned a unit, 3 units for A, obtaining an income of 1.9 units.

Figure 3.

Table 1. Stage I: amount allocation for Plant A

Amount s_1 available for Plant A	Decision d_1: amount to be allocated to Plant A in the stage:					Optimal reward
	0	1	2	3	4	
0	**0**					**0**
1	0	**0,2**				**0,2**
2	0	0,2	**0,5**			**0,5**
3	0	0,2	0,5	**1,5**		**1,5**
4	0	0,2	0,5	**1,5**	1,4	**1,5**

Table 2. Calculation of s_2

Amount s_2 available for Plants A and B	Decision d_2: amount to be allocated to Plant B; rest to be allocated to Plant A in optimal manner					Total optimal reward
	0	1	2	3	4	
0	**0**					**0**
1	0+0,2=0,2	0,3+0=**0,3**				**0,3**
2	0+0,5=0,5	0,3+0,2=0,5	0,6+0=**0,6**			**0,6**
3	0+1,5=**1,5**	0,3+0,5=0,8	0,6+0,2=0,8	1,2+0=1,2		**1,5**
4	0+1,5=1,5	0,3+1,5=**1,8**	0,6+0,5=1,1	1,2+0,2=1,4	1,5+0=1,5	**1,8**

Table 3. Detailed calculation of the stage s_2=3

d_2: allocation for B	Rest to allocate for A	Immediate reward	Optimum from stage I	Total reward	Total optimal reward
0	**3**	**0**	**1,5**	**1,5**	←
1	2	0,3	0,5	0,8	
2	1	0,6	0,2	0,8	
3	0	1,2	0	1,2	

Table 4. Stage III: allocation for Plant C; stage s_3=4

s_3: Amount available for Plants A, B and C	d_3: amount to be allocated to Plant C					Total optimal reward
	0	1	2	3	4	
4	0+1,8=1,8	0,4+1,5=**1,9**	0,9+0,6=1,5	1,1+0,3=1,4	1,6+0=1,6	**1,9**

Table 5. Stage III: *stage s_3=4*

Alternatives	Reward for plant C	Reward for best allocation between A and B (calculated in stage II)	Total optimal reward	
d_3=4 for C, 0 for A şi B	1,6	0	1,6	
d_3=3 for C, 1 for A şi B	1,1	0,3 (1 for B)	1,4	
d_3=2 for C, 2 for A şi B	0,9	0,6 (2 for B)	1,5	Max
d_3=1 for C, 3 for A şi B	**0,4**	**1,5 (3 for A)**	**1,9**	←

d_3=0 for C, 4 for A şi B	0	1,8 (1 for B, 3 pt A)	1,8	

Conclusion: Advantages of dynamic programming and its features

1. For each value of s at each stage, optimal income is calculated in the analysis made.
2. Marginal income for a given allocation policy as s increases (decreases) in every 1 million units can be easily seen from the calculations in the tables above.
3. One can easily perform a sensitivity analysis. For example, when taking into account only A and B, the best solution can be read from table 2 as: d_2=1 and d_1=3 for total revenue of $ 1.8 million.
4. Dynamic programming process identified and the second best alternative. In this case, in the Table 2 this is: d_3=0; i.e. by not allocating any amount for C, 1 for B and 3 for A, we'll obtain a $ 1.8 million profit. Similarly it can be obtained the third best solution, etc. These solutions are often important when qualitative factors must be taken into account.
5. Adding a new plant in the wording of the problem, to solve it will require only one additional stage calculations.
6. By adding money to be allocated for solving the problem will imply the occurrence of several states.
7. Problem-solving process by dynamic programming needs of 18 calculations. A complete solution given by enumeration would need only 15 calculations. Again, no one can save computational effort in such small problems. However, economy of effort is felt directly proportional with the increase in inputs of the problem.

BIBLIOGRAPHY

1. Bell P., C., Newson E., F.- *Statistics for business,* The Scientific Press, San Francisco, 1992
2. Boldur Gh., Raţiu-Suciu Camelia, Ciobanu Gh., Stancu I. – *Cercetare operaţională cu aplicaţii în economie,* ASE, Bucureşti, 1975
3. Boldur-Lăţescu, Gh., Săcuiu I., Ţigănescu E. – *Cercetarea operaţională cu aplicaţii în economie*, E. D. P., Bucureşti, 1979
4. Cazan E. – *Managementul producţiei,* Ed. Eurobit, Timişoara, 2000
5. Dinescu C., Săvulescu B.- *Metode matematice şi aplicaţii pentru optimizarea problemelor din economie,* E. D. P., Bucureşti, 1976
6. Dinescu C., Săvulescu B. – *Metode de matematică modernă pentru economie. Culegere de probleme,* E. D. P., Bucureşti, 1975
7. Leontief W. – *Analiza input-output,* Ed. Ştiinţifică, Bucureşti, 1970
8. Mendenhall W., Reinmuth J., E., Beaver R., Duhan D. – *Statistics for management and economics,* Duxbury Press, Boston, 1986
9. Popa H., L., Străuţi G., Vasu M., Pater F. – *Managementul şi ingineria sistemelor de producţie,* Ed. Politehnica, Timişoara, 2001
10. Postelnicu T., Dinescu C., Săvulescu B. – *Matematici speciale aplicate în economie,* E. D. P., Bucureşti, 1977
11. Rafiroiu M. – *Modele ale cercetării operaţionale aplicate în construcţii,* Ed. Tehnică, Bucureşti, 1980
12. Raţiu-Suciu C. – *Modelarea şi simulare proceselor economice,* E. D. P., Bucureşti, 1997
13. Turban E., Meredith J. – *Fundamentals of Management Science,* Irwin, Boston, 1991