JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 3, 358—360 (1961)

On the Reduction of Dimensionality for Classes of Dynamic Programming Processes

RICHARD BELLMAN

The RAND Corporation, Santa Monica, California

I. INTRODUCTION

The basic functional equation of dynamic programming has the form

$$f_n(p) = \max_q \,[R(p, q) + f_{n-1}(T(p, q))], \tag{1.1}$$

where n is a timelike variable. Generally, if numerical results are required for applications, digital computers must be used to obtain computational solutions. If p is a one- or two-dimensional vector, there is no difficulty in using the foregoing equation to obtain accurate solutions in a straightforward way and rapidly. If the dimension of p exceeds two, we run into the bottleneck of limited fast memory, and we must devise special techniques to overcome this difficulty. In our books [1, 2, 3], we have discussed a number of the special techniques that can be employed.

In this paper, we wish to present a new technique which appears quite promising. We shall briefly indicate its applicability in a method of successive approximations applied to processes of very high dimension, and to the digital computer determination of optimal play of chess and checkers. Detailed accounts of these matters, and of the application of this method to the computational solution of feedback control and trajectory problems will appear subsequently.

II. RELATIVE STATE

To reduce the dimensionality of a process, we introduce the concept of *relative state*. Taking a particular state p as a ground state or reference state, we determine the subsequent states of the system not in absolute terms, but rather in terms of the deviation from the reference state p. If the system can change in only a small number of ways at each stage, this description substantially reduces the number of values which must be tabulated in order to specify the state of the process at any subsequent time.

When the set of deviations itself becomes large enough to overflow the bounds of fast memory, we change the reference state to a new reference state, and begin over again. In this way, we can carry through a calculation involving a large number of stages.

We modify the equation of (1.1) by introducing the function $f_n(\delta) = f_n(\delta, p)$, where δ is the deviation from the fixed initial state p. The new equation is

$$f_n(\delta) = \max_q \left[R(\delta, q) + f_{n-1}(S(\delta, q))\right]. \tag{2.1}$$

If the set of deviations increases and becomes too large, we can introduce a new reference state every k stages. In order to avoid a confusion of notation, we can use a double subscript notation, $f_{n,m}(\delta)$, $n = 0, 1, \ldots, N$, $m = 0, 1, \ldots, k-1$, where n represents the number of stages remaining and m denotes the number of times we have shifted the reference state. Starting at p with an N-stage process, we obtain the equations

$$f_{n,0}(\delta) = \max_q \left[R(\delta, q) + f_{n-1,0}(S(\delta, q))\right] \tag{2.2}$$

for

$$n = N, N-1, \ldots, N-k+1,$$

$$f_{n,1}(\delta) = \max_q \left[R(\delta, q) + f_{n-1,1}(S(\delta, q))\right] \tag{2.3}$$

for

$$n = N-k, N-k+1, \ldots, N-2k+1,$$

and so on.

Observe that we are piecing together a solution of the general problem in a way which is quite different from that usually followed.

III. Successive Approximations

An immediate application of the foregoing ideas is to control processes of large dimension. As an approximation to optimal control, let us assume that the time intervals are taken so short that it is reasonable to assume that only one component changes per stage, and that these components change cyclically. The deviation of the state vector can now be expressed simply by means of a scalar quantity, and the technique discussed above can be employed.

IV. Checkers and Chess

Interesting examples of processes in which the set of all possible stater of the system is indescribably huge, but where the deviations are reasonably small in number, are checkers and chess. In checkers, the number of possible moves in any given situation is so small that we can confidently expect a complete digital computer solution to the problem of optimal play in this game. In chess, the general situation is still rather complex, but we can use the method described above to analyze completely all pawn-king endings, and probably all endings involving a minor piece and pawns. Whether or not this is desirable is another matter.

References

1. Bellman, R. "Dynamic Programming." Princeton Univ. Press, Princeton, New Jersey, 1957.
2. Bellman, R. "Adaptive Control Processes: A Guided Tour." Princeton Univ. Press, Princeton, New Jersey, 1961.
3. Bellman, R., and Dreyfus, S. "Computational Aspects of Dynamic Programming." Princeton Univ. Press, Princeton, New Jersey, 1962.