subparts are known as *fields*. Since fields may themselves refer to other records, complex structures can thus be developed. Many ramifications of the record concept are explored, and the application of records in different programming languages is discussed. An appendix specifies the necessary additions to ALGOL 60 in order to include records. Although by this time Hoare's concepts are quite well known, the article is still well worth reading and helps to provide a unifying framework for a number of related approaches to record handling.

The final article, by Ole-Johan Dahl, is a survey of discrete event simulation languages. Considering that Dahl is one of the authors of SIMULA, a leading simulation language, this is a remarkably even-handed and impartial discussion of the field. The author considers five well-known simulation languages: GPSS, SIMSCRIPT, CSL, SOL, and SIMULA. Examples are drawn from all of them. Dahl discusses the peculiar requirements of simulation languages, and takes pains to warn the reader of the hazards of making predictions from computer simulations. This article is in much the same spirit as the Hoare article, and develops a unifying framework in which the different languages can be viewed. There are interesting interrelationships between simulation languages and Hoare's record concept, since simulated entities usually have just the kind of structure that Hoare is concerned with. However, since the articles are separately written, this connection unfortunately is not made explicit.

PAUL ABRAHAMS

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

19[13.35].—ROBERT E. LARSON, *State Increment Dynamic Programming*, American Elsevier Publishing Co., Inc., New York, New York, 1968, xvi + 256 pp., 24 cm. Price $14.50.

Of all the primary ideas in optimization theory, dynamic programming has perhaps the most immediate and intuitive appeal. So fecund of application is "the principle of optimality" that one can readily forgive its ignorance of the subjunctive for the sake of the constant challenge it presents to the ingenuity in adapting it to a vast range of situations. Its basic notion can be seen from the simplest of examples, yet it reaches to the most recondite problems. It is in fact—as I once heard Bellman remark—"just mathematics; not difficult; all it requires is intelligence."

The two main drawbacks of dynamic programming were its lack of precision and what Bellman colorfully called "the curse of dimensionality." The work of Berkovitz and Dreyfus in 1964 laid a rigorous foundation and permitted the proper conditions of continuity to be imposed on optimal solutions. At about the same time Larson was beginning to overcome the practical difficulties that arise with more than one or two state variables. It is this work that he has now most usefully presented in book form. State increment dynamic programming starts from the standard functional equation of dynamic programming, but uses two ingenious modifications to reduce the computing time and storage requirements. The first is to choose the time-like increment so as to keep each successive state within a prescribed hypercube centered on the previous one. The second is to carry out the

computations by blocks rather than in the sequence of time-like increments. The first saves storage and time in the high speed memory; the second saves time in transferring between low speed and high speed storage.

After an outline of the conventional procedure, the book gives a very clear description of the state increment computational procedure with various useful modifications. Many interesting examples are given in detail and a final chapter gives a development of the method of successive approximations. The author is to be congratulated on a notable and well-presented contribution to the art and craft of dynamic programming.

RUTHERFORD ARIS

Department of Chemical Engineering
University of Minnesota
Minneapolis, Minnesota 55455