# An Algorithm for Path Connections and Its Applications*

C. Y. LEE†, MEMBER, IRE

*Summary*—The algorithm described in this paper is the outcome of an endeavor to answer the following question: Is it possible to find procedures which would enable a computer to solve efficiently path-connection problems inherent in logical drawing, wiring diagramming, and optimal route finding? The results are highly encouraging. Within our framework, we are able to solve the following types of problems:

1) To find a path between two points so that it crosses the least number of existing paths.
2) To find a path between two points so that it avoids as much as possible preset obstacles such as edges.
3) To find a path between two points so that the path is optimal with respect to several properties; for example, a path which is not only one of those which cross the fewest number of existing paths, but, among these, is also one of the shortest.

The minimal-distance solution has been programmed on an IBM 704 computer, and a number of illustrations are presented. The class of problems solvable by our algorithm is given in a theorem in Section III. A byproduct of this algorithm is a somewhat remote, but unexpected, relation to physical optics. This is discussed in Section VI.

## I. Introduction

In processing information consisting of patterns, rather than numbers or symbols, on a digital computer, we may wish to know how a computer, without sight and hearing, can be made to deal competently with situations which appear to require coordination, insight, and perhaps intuition. It is not our intention to consider the general problem of pattern detection and recognition by machines. We will consider rather the following simpler, and therefore perhaps more basic, problem in pattern processing by machines.

Let a pattern of some sort be presented to a machine. We then want the machine to construct some optimal path subject to various constraints imposed by the pattern. The problem is to find efficient procedures, which, if followed by the machine, would lead to an optimal solution.

Ideally, many situations would fall within this description. We might present to the machine a map of Manhattan and ask it to find the shortest-time route between, say, the United Nations and Yankee Stadium, using only public transportation. With sufficient care, it is possible to make a problem such as this unambiguous. In most cases, however, it would be too great a struggle just to present the problem in a way that is completely and consistently stated. For this reason, we have decided to present an abstract model in Section II. Based on this model, we will consider a class of well-defined optimal path problems. A general procedure for solving this class of problems will then be given in Section III.

Within this class of problems is the *shortest-route problem* on which there has been earlier definitive work. Algorithms for finding shortest paths have been given by Dantzig,[1] Ford and Fulkerson,[2] Moore[3] and Prim.[4] The minimal-distance illustrations (Section V) make use of one of Moore's algorithms, which is a specialization of algorithm A given in Section III. These experiments were tried out before the abstract model was completed. Once we have at our disposal the abstract model, it came as a pleasant surprise that problems such as the minimal-crossing and minimal edge-effect problems, which had appeared difficult to us previously, all yielded immediately to algorithm A. The possibility of joint minimization is also a direct consequence of algorithm A. A further outcome was the "diffraction" patterns. These experiments would not have been attempted if we had not noticed the patterns obtained in the minimal-distance experiments.

## II. An Abstract Model and the Path Problem

### A. $\mathcal{C}$-Space, an Abstract Model

Let $C$ be a set of elements called cells: $C = \{c^1, c^2, \cdots\}$. For each cell $c^i$ in $C$, there is defined a subset of $C$ called a *1-neighborhood* $N(c^i)$ of $c^i$: $N(c^i) = \{c_1{}^i, c_2{}^i, \cdots, c_n{}^i\}$. The following rules hold for 1 neighborhoods:

N1) Every 1-neighborhood has in it exactly $n$ cells, where $n$, $n \geq 1$, is some predetermined number depending on the specific model involved.

N2) If $c^i \in N(c^i)$, then $c^i \in N(c^i)$. We will call the function $N$ with domain $C$ and range subsets of $C$ the *1-neighborhood function.*

Together with the 1-neighborhood functions, there are $n$ functions $d_1, d_2, \cdots, d_n$ on $C$ to $C$ defined as follows: If $c^i$ is any cell in $C$ and $N(c^i) = \{c_1{}^i, c_2{}^i, \cdots, c_n{}^i\}$, then

$$d_k(c^i) = c_k{}^i, \qquad k = 1, 2, \cdots, n.$$

* Received by the PGEC, December 2, 1960. This material was presented as part of the University of Michigan Engineering Summer Session, Ann Arbor, June 19–30, 1961.

† Bell Telephone Labs., Inc., Whippany, N. J.

[1] G. B. Dantzig, "Maximization of a Linear Function of Variables Subject to Linear Inequalities," Cowles Commission; 1951.

[2] L. R. Ford and D. R. Fulkerson, "Maximal flow through a network," *Can. J. Math.*, vol. 8, pp. 399–404; 1956.

[3] E. F. Moore, "Shortest path through a maze," in "Annals of the Computation Laboratory of Harvard University," Harvard University Press, Cambridge, Mass., vol. 30, pp. 285–292; 1959.

[4] R. C. Prim, "Shortest connection networks and some generalizations," *Bell Sys. Tech. J.*, vol. 36, pp. 1389–1401; November, 1957.