Soft Architecture Machines

OA Funding Provided By:

National Endowment for the Humanities/Andrew W. Mellon Foundation Humanities
Open Book Program.

The title-level DOI for this work is:

doi:10.7551/mitpress/6317.001.0001

Epilogue

An Allegory

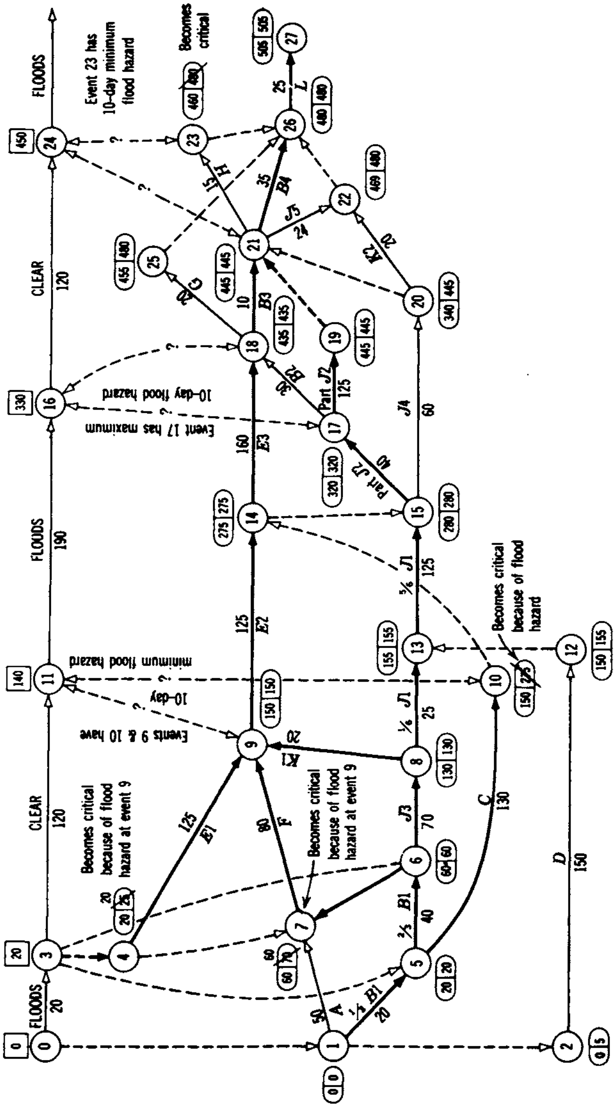
I have heard versions of the following story on several occasions, and I have told varying versions on many occasions. For these reasons I am no longer sure where I first heard it or of its original form or from whom (though I tend to think it was Seymour Papert). The story, nonetheless, has close analogies in the historical development of architecture as assisted by computers. The story is about a machine. It is called the string-and-ring machine.

There exists a classic combinatorial problem in mathematics called the *traveling salesman problem*. It considers *N* geographically distributed locations interconnected by "roads." The problem is to find the shortest route that will take a salesman to every city with the shortest possible mileage without going through any city twice. Note that the problem has important practical applicatons in the routing of pipes, wires, and communications networks. Consequently it has been studied at great length (Bellmore and Nemhauser, 1968; Arnoff and Sengupta, 1961; Karg and Thompson, 1964; Dantzig, Fulkerson, and Johnson, 1959; Croes, 1958; Gomory, 1966; Flood, 1956; Heller, 1955; Little, Murty, Sweeney, and Karel, 1963; Lambert, 1960; Morton and Land, 1955; Roberts and Flores, 1966; Raymond, 1969; Wootton, 1969; Srinastava et al., 1969; Rothkopf, 1966). "Although some ways have been found for cutting down the length of the search, no algorithm has been discovered sufficiently powerful to solve the traveling salesman problem with a tolerable amount of computing for a set of, say, fifty cities" (Simon, 1970). Consider that the number of alternative routes is $N-1$ factorial (which for fifty cities is greater than 3×10^{64}).

Another version of the problem, equally well studied (Beardwood, Halton, and Hammersky, 1959; Dantzig, 1960; Butas, 1968; Dreyfus, 1969; Hu, 1968; Hoffman and Markowitz, 1963; Hu and Torres, 1969; Nicholson, 1966; Mills, 1968 and 1966; Pollack and Wiebenson, 1960; Peart, Randolph, and Bartlett, 1960; Verblunsky, 1951), is to find the shortest path from one given point on the network to another given point. It is the history of this particular version of the traveling salesman problem (usually referred to as the shortest path problem) that I wish to break into "generations."

The first era is the obvious application of a machine to a task unmanageable by a human and is characterized by an exhaustive search for all possible solutions. Note that this method does yield the optimum solution, because all alternatives are searched (and there happens to be only one goal, shortness of path). This was the era of exhaustive searching.

The second era of approach to the problem is characterized by the following attitude: Let the machine do what it is good at doing, let the man do what he is good at doing, and provide the two with a smooth interface such that they can work effectively. Hence, a typical solution would be to display on a cathode-ray tube the map of *N* cities and have the human operator of the console point at a "reasonable" set of nodes that lie between *A* and *B*. The machine's task is simply to sum up the mileages and display the total. Continuing, we allow the user to alter his routing interactively so that as he moves the line of travel he receives a constant updating of the new mileage. In this manner he can "massage" the route and within a short period of time come up with a "very good" route (conceivably the optimum).



A typical critical path method used in construction practice. The particular example is of the construction of a rock fill dam taken from *Critical Path Methods in Construction Practice*, Antill and Woodhead, New York: Wiley & Sons, 1965. Note that, to the chagrin of CPM enthusiasts the string and ring machine cannot be run backward. That is, it cannot compute the longest trip distance.

The third era of the problem is characterized by wondering just what the human was bringing to the problem that the machine could not possess itself: what pattern-recognition abilities and, particularly, what heuristics? Hence, the approach of the third era was to develop heuristics that could limit the search, reducing the alternatives to a few thousand or even a few hundred reasonable ones. We can imagine such rules of thumb as: It is probably not worth backtracking for more than a certain percentage of the total distance; the route probably lies within a certain subset of the map, as described by an upper left and lower right, for example; look for roads that tend to be straight; and so on.

The fourth era is that of a special-purpose machine. It is composed of *N* labeled shower curtain rings interconnected to each other with kite string of a length proportional to the actual road distance between the cities. Once constructed, this computing mechanism can be employed by simply picking up the two rings that represent the two cities in question, by pulling, and by observing which strings become taut first. We have the optimum route generated by a machine. We call it the string-and-ring machine.

I tell this long story, not because I believe necessarily that there is a string-and-ring machine for architecture, but because I see a similar historical development. The first applications of computers to architecture were quite similarly characterized, as in era one, by exhaustive searching. The approach and attitude were to make the problem simple enough to examine all solutions in order to post the best. This approach has proved quite useless in all cases except the most belittling exercise and hence receives little further study.

The second era of computer-aided architecture has been the "partition paradigm": let the designer do what he is good at and let the machine do what it is good at, and so forth. Of course, computer graphics bolstered this approach and assisted in affording the requisite smooth interface. My own earlier work on URBAN5 can be considered exemplary of this approach, and it did not work. It did not work because no matter how many trinkets and how much paraphernalia the interface had, the machine still could not contribute to finding answers (and finding questions) because it did not understand! It could not handle missing information, context, and so on; and it was always at the mercy of the validity of its inputs (and me).

The third era is maybe where we are now. We are trying to understand just what the human does bring to the design process and, at the same time, who that human should be. What heuristics do we use, and how do we use them? Are some people innately better designers than others? If so, why? Questions like these characterize our present efforts. I believe that I can use "our" much more broadly than the polemics of this volume may suggest.

And maybe there is a string-and-ring machine for architecture.