A NEW APPROACH TO THE PLACEMENT PROBLEM

1970

SovietRxiv

Abstract

Full Text

UDC 519.14

CYBERNETICS AND CONTROL THEORY

Yu. L. ZIMAN

A NEW APPROACH TO THE PLACEMENT PROBLEM

(Presented by Academician S. A. Lebedev, 21 V 1970)

Among the problems of integer programming, the problem of locating production units, or the placement problem, is well known ([1]). In this problem one is given the number of objects n, which must be placed in $n' \geq n$ possible locations, as well as the distances d_{ij} from location i to location j $(i, j = 1, 2, \dots, n')$ and the flows f_{kl} between object k and object l $(k, l = 1, 2, \dots, n)$. These terms reflect generalized concepts and are introduced in order somehow to evaluate the placement of a particular pair of objects in two particular locations. Suppose that after the placement all occupied locations have been renumbered so that each location is assigned the number of the object placed in that location. Then the evaluation of the placement is the sum

$$L = \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{ij},$$

and the solution of the placement problem is a placement of all elements that minimizes the quantity L.

In such a formulation the problem is not difficult to reduce to a quadratic assignment problem. Since the flows are specified for pairs of objects, they can be represented in the form of a graph G with n vertices, in which any two vertices i and j corresponding to objects for which $f_{ij} \neq 0$ are joined by an edge with weight f_{ij}. For practical applications it is convenient to use the adjacency matrix of the graph G, in which the ones are replaced by the corresponding values of the flows.

Of all applications, we shall be interested primarily in the placement of the cells of an electronic unit on a plane, minimizing the total length of connections between the cells. In this case the flow graph is a graph of connections, and all f_{ij} are integers.

Let us consider a more general formulation of the problem, one that better corresponds to the engineering specification of connections. Let us call the n

objects to be connected *elements*. On the set N of all elements, specify a certain set of subsets $S_1, S_2, \dots, S_m$ of its elements such that $S_1 \cup S_2 \cup \dots \cup S_m = N$. Each S_i may include from one to n elements of N. We shall call the subsets S_i *connections*, and the number of elements in each connection the *order* of that connection. Suppose now that each connection includes those elements which must be connected with one another, and that the connection is implemented in the form of a tree on the elements of this connection. Thus, if the order of a connection is p, then there exist $p^{(p-2)}$ different ways of implementing the connections for this connection.

The set of elements N, together with the connections specified on it in the indicated way, will be called a *scheme*.* If for each connection a particular tree is chosen, then this scheme may generate a graph G_s, which we shall call the *schemograph* of this scheme. If the orders of the connections in the scheme are $p_1, p_2, \dots, p_m$, then such a scheme can generate

$$\prod_{i=1}^{m} p_i^{(p_i-2)}$$

different schemographs. Each schemo-

* In note (^{2}) the term "scheme" denoted another concept; in an unpublished work by A. M. Stepanov an analogous object is called a scheme, but nothing is said about the manner of connecting the elements of a connection.

the graph indicates one of the ways of carrying out the connections in the circuit. It is not difficult to see that it is a generalization of a graph without loops. Indeed, if the orders of all connections in the circuit do not exceed two, then the given circuit will have a single schematic graph, i.e., it degenerates into a graph whose edges are formed by connections with $p = 2$. A connection of order equal to one corresponds, if the element does not enter into any other connection, to an isolated vertex of such a graph.

In accordance with [2], we shall call a graph *proper* if it admits an embedding in the square integer lattice of points α such that the vertices lie at the nodes of the lattice, and the edges connect points adjacent horizontally, vertically, or diagonally. The embedding itself will also be called *proper*. We shall compute the distance d_{ij} between points i and j on the lattice α as the minimal number of segments of a polygonal line that can be constructed on α so that each segment connects adjacent points on α. It is not difficult to verify that this definition of distance satisfies the basic axioms of a metric and, consequently, defines on α a certain metric.

For a certain embedding of a graph G with m edges in the lattice α, compute the sum, which we shall call the *embedding defect*:

$$\Delta = \sum_{i=1}^{m} (d_i - 1).$$

Here d_i is the distance on α between the endpoints of the i-th edge, and the summation is over all edges of the graph G. It is easy to see that for a proper embedding $\Delta = 0$. Thus, the embedding defect gives a quantitative measure of the "improperness" of the embedding. Now we shall call a circuit *proper* if at least one schematic graph of it is proper. For a given embedding of the elements, we shall call the *defect* Δ_s *of embedding a circuit* in the lattice α the minimal defect over all schematic graphs of the circuit. An embedding of a circuit for which $\Delta_s = 0$ will accordingly be called a *proper embedding*.

In terms of a proper embedding, the placement problem can now be formulated, when the connections are specified by a circuit:

Find an embedding of the circuit S in the lattice α with minimal defect Δ_s.

Proceeding from such a statement of the problem, one can construct combinatorial algorithms, for example of sequential type, that seek an embedding of the given circuit with minimal defect. Naturally, the dimensions of the lattice α may be subject to necessary restrictions associated with the particular design.

Let us add also that circuits in general, and proper circuits in particular, constitute interesting objects for research as radical generalizations of graphs.

Institute of Precision Mechanics and Computer Engineering
Academy of Sciences of the USSR
Moscow

Received
7 V 1970

CITED LITERATURE

1. A. A. Korbut, Yu. Yu. Finkelstein, *Discrete Programming*, "Nauka," 1969.

2. Yu. L. Ziman, DAN, 162, No. 4 (1965).

Note: Figure translations are in progress. See original paper for figures.