The minimisation of distance in placement algorithms

V. A. LoDato

Simulating Science and Engineering, 3 Daisy Lane, Wappinger Falls, N.Y., USA

The criterion of total length in placement algorithms is examined by means of statistical inferences. It is shown that it is a poor global constraint and appears to be a good local constraint. This is verified by coupling a partitioning and placement algorithm and comparing it to a placement algorithm using total length as its only constraint.

(Received February, 1970)

The usual problem in the placement of components of a data processing machine is to assume a certain geometric figure on which the components are placed; and secondly, to place them in such a manner that the total length of interconnections is a minimum, the assignment algorithm (Mamelok, 1966). However, if placement is done with this constraint as a principal one, it is quite possible that wiring may not be achieved efficiently (Pomentale, 1965; Steinberg, 1961).

In this paper we look at the total length constraint and modify it by a free parameter, α, which affects the distance metric between vertices; this abstractly has the effect of introducing other criteria that certainly exist, due to technological rules and thereby effect an optimal placement. We also show that this criterion should not be *the* constraint in a placement algorithm on a global basis for linear and square arrays. The calculation of an optimal distribution of edges is given for certain values of α along with the properties of these distributions.

Based on the above, consider an algorithm is developed that does not perform placement on the total package; but initially partitions the package in such a manner as to enhance wireability and then an assignment algorithm is invoked on a *local* basis.

1. Theory

The theory of placement of components of any data processing machine can abstractly be represented as a linear (non-oriented) graph $G = [V(G); E(G)]$ where $V(G)$ is the set of vertices $\{V_R\}$ and $E(G)$ the set of edges. The set of edges can be represented as $E(G) \subseteq \{(V_i, V_j): V_i, V_j \in V(G)\}$ where $\{V_i, V_j\}$ denotes an unordered pair. Given graph G, G' is said to be a subgraph of G if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. The union $G_i \cup G_j$ and the intersection of $G_i \cap G_j$ for $(i \neq j)$ are defined as $G_i \cup G_j = \{E(G_i) \cup E(G_j); V(G_i) \cup V(G_j)\}$ and $G_i \cap G_j = \{E(G_i) \cap E(G_j); V(G_i) \cap V(G_j)\}$.

The graph that is used to represent the machine is a graph where each edge is distinguishable from any other and a pair of vertices may have any number of edges connecting it. We are concerned now with the average properties of the graph rather than the individual graphs. If $|G|$ is the cardinality of the graph with V vertices and E edges, then G is

$$G = [V(V-1)/2]^E \tag{1.1}$$

We now define the total distance of graph g with placement p and parameter α (where $\alpha \geq 0$) as

$$D_p^\alpha(g) = \sum_{i<j} e_{ij}(g)\, d^\alpha p_i p_j \tag{1.2}$$

where $e_{ij}(g)$ are the corresponding edges of graph g and $d^\alpha p_i p_j$ is the distance for placement p from the ith to the jth vertex raised to the α power. The average value for all graphs is defined as

$$D^\alpha(g) = \sum_{\{p\}} \frac{D_p^\alpha(g)}{\{p\}} \tag{1.3}$$

where the sum in (1.3) is carried out over all permutations. The minimum placement is

$$D_{\min}^\alpha(g) = \min_{\{p\}} [D_p^\alpha(g)] \tag{1.4}$$

Therefore, the average minimum distance in the sum over all minimum distances divided by the cardinality of the graph $|G|$ is,

$$D_{\min}^\alpha = \sum_{g \in G} \frac{D_{\min}^\alpha(g)}{|G|} \tag{1.5}$$

and it is also equal to

$$D_{\min}^\alpha = \sum_{g \in G} \frac{D^\alpha(g)}{|G|} = D^\alpha \tag{1.6}$$

This can easily be shown since

$$D^\alpha(g) = \sum_{\{p\}} \frac{D_p^\alpha(g)}{|\{p\}|} \tag{1.7}$$

Substitute (1.2) into the above we have

$$D^\alpha(g) = \frac{1}{|\{p\}|} \sum_{\{p\}} \sum_{i<j} e_{ij}(g)\, d^\alpha p_i p_j \tag{1.8}$$

or,

$$D^\alpha(g) = \sum_{i<j} e_{ij}(g) \frac{1}{|\{p\}|} \sum_{\{p\}} d^\alpha p_i p_j \tag{1.9}$$

Since the total number of edges is

$$E = \sum_{i<j} e_{ij}(g) \text{ and } \frac{1}{|\{p\}|} \sum_{\{p\}} d^\alpha p_i p_j = \frac{\sum_{i<j} d_{ij}^\alpha}{D(D-1)}$$

where D is the number of points in the array. Hence we have

$$D_{\min}^\alpha = D^\alpha$$

Q.E.D.

Let V_k be the number of pairs of points in a geometrical array (square, straight line, etc.) with array distance k. Let us also consider the set of placements with n_k edges. The number of placements p of a given class with V vertices and E edges is

$$P[\{e_k\}] = V!\,E! \prod_k \frac{V_k e_k}{e_k!} \tag{1.10}$$

Let us consider the number of placements of a given distance D^α and E edges as

$$P_D(\alpha)_E = \sum_{\{e_k\}} P[\{e_k\}] \tag{1.11}$$

where the distributions $\{e_k\}$ are constrained to satisfy

$$\sum_k e_k = E \tag{1.12}$$

$$\sum_k k^\alpha e_k = D^{(\alpha)} \tag{1.13}$$

A way to determine $P_D(\alpha)_E$ is to find the distribution $\{\tilde{e}_k\}$ which maximises $\ln \{P[M_R]\}$ for $\sum_k e_k = E$ and $\sum_k k^\alpha e_k = D^\alpha$.

For all cases of a straight line we have $V_k = W-k$ while for a square the total number of vertex parts of distance k possible is

$$V_k(k \leqq W) = 2kW(W-k)+k(k^2-1)/3 \tag{1.14*}$$

Now using (1.12) and (1.13) as non-holonomic constraints we are able, by using Lagrange multipliers, to maximise (1.10) and we obtain

$$\tilde{e}_k = V_k e^{\lambda+\mu k^\alpha} \tag{1.15}$$

It is now possible to evaluate λ and μ by using (1.12) and (1.13)

$$\frac{\sum V_k k^\alpha e^{-\mu k^\alpha}}{\sum V_k e^{-\mu k^\alpha}} = R^\alpha \tag{1.16}$$

For given values of R and α we evaluate a value of μ (R^α) and substituting this value into (1.15) and (1.13) and find the corresponding λ (R^α). Thus, we are able to find $P[\tilde{e}_k]$. Since,

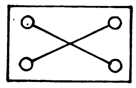
$$P[\{\tilde{e}_k\}] = \exp\left[-\sum \tilde{e}_k \ln \frac{\tilde{e}_k}{V_k} + N \ln N + V \ln \frac{e}{V}\right] \tag{1.17}$$

Using this formulation we are able to calculate the distribution $P[\{e_k\}]$.

2. Conclusion

As the size of the graph increases the average distance of an edge for a minimum placement also increases; also as α increases the average edge distance increases. This is what one might expect. However, the variance of each graph is small and as the number of edges increases the variance decreases. It should be noted that the variance of a square is greater than that for a linear array. However, as the number of edges starts to increase, the distribution becomes like a delta function. This means that for an optimal placement there is a very small spread in the number of different size edges; and from skewness this spread is symmetric.

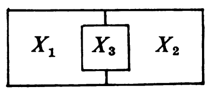
It should be kept in mind that the major reason for placement is to make the package wireable. Partitioning done iteratively (to the unit cell) on a geometrical array is placement. In order to minimise serious interconnections of the form:



partitioning is employed before an assignment algorithm is used. The criterion of minimisation of the total length does not necessarily decrease the serious interconnections, and thus may hinder wireability.

Given a set of logic blocks, and input/output connections we wish to arrange them in an optimal position on a square array. In our algorithm the I/O units can only be placed on the perimeter of a square while the logic blocks can be placed anywhere in the interior. The logic blocks are first assigned and then the I/O are assigned. In the placement of the logic blocks, one first constructs the total graph for the interconnection between blocks. The graph $G = (X,\Gamma)$ is first decomposed into three components $\{X_1, X_2, X_3\}$ such that the interconnections between X_2 and X_3 are the null set; $X_1 \cap \Gamma(X_2) = \phi$, $X_2 \cap \Gamma(X_1) = \phi$.

All the interconnections are via X_3.



If the set is a minimum the set of vertices in X_3 are the points of articulation of the graph.

In order to find the points of articulation, let the boolean matrix of the graph G be denoted by $[N]$ and the complementary matrix $[\bar{N}] = 1 - [N]$. The submatrix of $[\bar{N}]$ is defined by the rows corresponding to X_1, and the columns corresponding to X_2, has all its elements equal to unity in accordance with the relations $X_1 \cap \Gamma(X_2) = \phi$ and $X_2 \cap \Gamma(X_1)$

*Where W is the dimension of the array.

Table 1—Linear array

α	VERTICES	EDGES	MEAN	VARIANCE	SKEWNESS	KURTOSIS
0·5	10	50	3·34	8·94 (−02)*	3·08 (−03)	2·64 (−02)
1·0	10	50	3·68	9·43 (−02)	1·51 (−03)	2·83 (−02)
1·5	10	50	3·99	9·89 (−02)	7·48 (−04)	2·64 (−02)
2·0	10	50	4·28	1·02 (−02)	−2·29 (−04)	2·22 (−02)
0·5	20	50	6·23	4·00 (−01)	3·88 (−02)	4·57 (−01)
1·0	20	50	7·03	4·15 (−01)	2·27 (−02)	5·23 (−01)
1·5	20	50	7·73	4·29 (−01)	6·10 (−03)	5·62 (−01)
2·0	20	50	8·35	4·42 (−01)	−7·81 (−03)	5·66 (−01)
0·5	10	100	3·33	4·46 (−02)	5·18 (−04)	8·54 (−03)
1·0	10	100	3·67	4·74 (−02)	1·22 (−04)	9·77 (−03)
1·5	10	100	3·99	4·96 (−02)	−2·90 (−04)	1·12 (−02)
2·0	10	100	4·28	5·13 (−02)	−1·37 (−04)	3·66 (−03)
0·5	20	100	6·21	2·00 (−01)	1·15 (−02)	1·21 (−01)
1·0	20	100	7·01	2·08 (−01)	4·15 (−03)	1·56 (−01)
1·5	20	100	7·73	2·15 (−01)	1·22 (−03)	1·64 (−01)
2·0	20	100	8·36	2·21 (−01)	−2·68 (−03)	1·52 (−01)

*8.92 (−02) = 8.92 x 10^{-2}

Table 2—Square array

α	VERTICES	EDGES	MEAN	VARIANCE	SKEWNESS	KURTOSIS
0·5	25	20	2·85	8·32 (−02)	1·50 (−03)	2·05 (−02)
1·0	25	20	2·49	7·84 (−02)	5·04 (−04)	1·57 (−02)
1·5	25	20	3·12	7·41 (−02)	−7·93 (−04)	1·46 (−02)
2·0	25	20	3·24	7·10 (−02)	−1·56 (−04)	1·27 (−02)
0·5	25	30	2·85	5·45 (−02)	4·12 (−04)	9·89 (−03)
1·0	25	30	2·99	5·12 (−02)	−1·53 (−04)	8·50 (−03)
1·5	25	30	3·13	4·77 (−02)	−7·02 (−04)	8·06 (−03)
2·0	25	30	3·24	4·57 (−02)	−1·27 (−03)	8·30 (−03)
0·5	100	20	5·47	3·33 (−01)	8·54 (−03)	2·79 (−01)
1·0	100	20	5·78	2·99 (−01)	1·22 (−03)	2·16 (−01)
1·5	100	20	6·04	2·78 (−01)	−4·39 (−03)	1·80 (−01)
2·0	100	20	6·28	2·60 (−01)	−9·52 (−03)	1·75 (−01)
0·5	100	30	5·46	2·01 (−01)	−4·88 (−04)	1·21 (−01)
1·0	100	30	5·78	1·82 (−01)	−2·93 (−03)	9·83 (−02)
1·5	100	30	6·05	1·70 (−01)	−4·63 (−03)	9·28 (−02)
2·0	100	30	6·29	1·60 (−01)	−6·10 (−03)	8·98 (−02)

$= \phi$. The complete submatrix is not a submatrix of any other complete submatrix for if it were $X_3 = X - X_1 \cup X_2$ where $X = \{X_1, X_2, X_3\}$ would not be minimal.

Hence all we need to find are all the complete submatrices of $[\bar{N}]$ (primary matrices) which define two subsets X_1 and X_2.

To obtain the primary matrices the Malgrange algorithm was employed (Kaufmann, 1967). Since we are able to find the primary matrices the points of articulation are generated X_3.

Based on the set of vertices in X_3, X_1 and X_2 are decomposed into X_{11}, X_{21} and X_{12}, X_{22} such that the interconnections between X_{11} and X_{22} is the null set also the interconnections between X_{21} and X_{12} is the null set.

X_{11}	X_{12}
X_{21}	X_{22}

After the subgraphs X_1 and X_2 are decomposed, the subgraph X_3 is annihilated and its vertices are placed in any one of the subgraphs, however with the constraint that no serious interconnections are created. The evaluation function for the decomposition of X_1 and X_2 is based on probabilistic arguments, for example the seeds for X_{11}, X_{21} are chosen to be the two most 'mutually' disjoint vertices with the most interconnection. Obviously the seeds may not be unique; in this case one is chosen at random from the subset of seeds. In most logic blocks one will find two-four edges emanating from each vertex. Hence, if we have 25 vertices, the density of non-zero elements in the boolean matrix is 0·125. This says that the matrix is sparse and thus the graph is not strongly connected, and that evaluation functions need not be all exhaustive to obtain in general good results.

Now that the logic blocks are partitioned the I/O pins are assigned with the only constraint that serious interconnections are not created, if so then a logic block or blocks are moved so that serious interconnections do not evolve.

It is at this stage that an assignment algorithm is employed. For a complete detail description of the algorithm the reader is referred to the following reference (Hoffman and Markowitz, 1963). This algorithm is employed on each respective component of the graph. Thus, we have employed minimisation of the total length as a secondary criterion.

A number of 25 × 25 arrays have been studied and it has been found that using a partitioning algorithm coupled with a placement algorithm, and using the distance criterion as a local constraint; wireability was improved by 10%.

References

Hoffman, A. J., and Markowitz, H. M. (1963). A Note on the Shortest Path, Assignment, and Transportation Problem, *Naval Research Logistics Quarterly*, Vol. 10, No. 4.

Kaufmann, A. (1967). *Graphs, Dynamic Programming and Finite Games*, Academic Press, New York.

Mamelok, J. S. (1966). The Placement of Computer Logic Modules, *JACM*, Vol. 13, pp. 615-629.

Pomentale, T. (1965). An Algorithm for Minimizing Backboard Wiring Functions, *CACM*, Vol. 8, pp. 699-703.

Steinberg, L. (1961). The Backboard Wiring Problem: A Placement Algorithm, *SIAM Rev.*, Vol. 3, pp. 37-50.