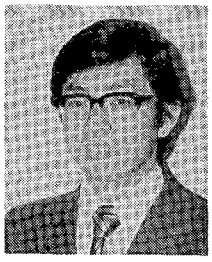


Shuji Tsukiyama (S'76–M'77) was born in Osaka, Japan, on November 23, 1949. He received the B.S., M.S., and Ph.D. degrees all in electronic engineering from the Faculty of Engineering, Osaka University, Suita, Osaka, in 1972, 1974, and 1977, respectively.

He has been with the Department of Electronic Engineering, Osaka University, as a Research Assistant. During the academic year of 1978–1979, he was with the Electronics Research Laboratory, University of California, Berkeley, as a Visiting Research Engineer. His main interest is in graph theory, computational complexity, and computer-aided design.

Dr. Tsukiyama is a member of the Institute of Electronic and Communication Engineers of Japan.

Isao Shirakawa (S'62–M'68) was born in Toyama Prefecture, Japan, on September 12, 1939. He received the B.S., M.S., and Ph.D. degrees, all in electronic engineering from the Faculty of Engineering, Osaka University, Suita, Osaka, Japan, in 1963, 1965, and 1968, respectively.

Since 1968 he has been with the Department of Electronic Engineering at the same university, where he is now an Associate Professor. During the academic year of 1974–1975 he was with the Electronics Research Laboratory, University of California, Berkeley, as a Visiting Research Engineer. His current interest is mainly in applied graph theory, design automation, and computer-aided design associated with layout problems.

Dr. Shirakawa is a member of the Institute of Electronic and Communication Engineers of Japan and Information Processing Society of Japan.

Shigeo Asahara was born in Osaka, Japan, on September 17, 1953. He received the B.S. and M.S. degrees in electronic engineering from Osaka University, Osaka, Japan, in 1976 and 1978, respectively.

He is currently a Graduate Student working towards the Ph.D. degree at the same university. His main interest is in layout for PWB's and LSI's.

A Forced Directed Component Placement Procedure for Printed Circuit Boards

NEIL R. QUINN, JR., AND MELVIN A. BREUER, SENIOR MEMBER, IEEE

Abstract—**This paper deals with the problem of placing components on a carrier, such as a printed circuit board (PCB). We present a new mathematical formulation of the concept of force directed placement, and describe an efficient computational procedure for solving the resulting system of equations. The placement procedure is broken down into two phases, Phase I being the "relative location phase," and Phase II being the "slot assignment or component overlap resolution phase." In Phase I of the procedure, we solve a set of simultaneous equations, based upon the interconnection topology of the system of components, in an endeavor to determine the optimum relative location of every component with respect to every other component. The equations are set up such that there are attractive forces between components sharing a common signal, and repulsive forces between components having no signals in common. The results of Phase I are often unacceptable from a physical standpoint because there is a great deal of overlap among the components. Phase II eliminates component overlap by either of two methods, depending upon the physical properties of the carrier. If the carrier is subdivided into slots, then the components are assigned to these slots using a criteria which minimizes the total distance that all components need be moved. We perform this assignment by using the linear assignment algorithm. If the carrier is such that components are allowed to reside anywhere, then a different technique to resolve component overlap is used.**

A parametric analysis of the procedure is given based upon 12 different PCB's. These results show comparisons of this method to the work of others, and provide some insight into the method's absolute merits.

Keywords: **Assignment, placement, layout, printed circuit boards, force-directed placement.**

I. The Placement Problem

THIS PAPER deals with the problem of placing components on a carrier, such as a printed circuit board (PCB) or ceramic substrate. Hanan and Kurtzberg [5] state, "in the real world, there are actually a number of

Manuscript received April 20, 1978. This work was supported in part by the National Science Foundation under Grant ENG74-18647.

N. R. Quinn, Jr., is with Environmental Management Systems, Inc., San Diego, CA.

M. A. Breuer is with the University of Southern California, Los Angeles, CA.

0098-4094/79/0600-0377$00.75

(possibly conflicting) goals which must be satisfied in a placement configuration: wire buildup in the routing channels must be kept within tolerable bounds; signal cross-talk must be eliminated without undue use of expensive shielding techniques; signal echoes must be eliminated; heat dissipation levels must be perserved, etc."

It is clearly impractical, if not impossible, to directly incorporate into a placement algorithm all of the real world goals mentioned above. However, minimizing the total weighted wire length of the circuit is one criteria that can be used to partially satisfy these goals en masse, and is the usual measure adopted for placement algorithms. We shall also use this objective.

A. Previous Placement Techniques

There are basically two different categories of placement problems, referred to as the continuous and noncontinuous (array) problems. For the *continuous case*, the placement media, such as a PCB, can be treated as a continuous plane on which the components to be placed are free to reside. In the *noncontinuous case*, the media is partitioned into an array of slots into which the components are placed (assigned).

A fairly complete discussion of the noncontinuous case can be found in Breuer [7]. The major contributors to the continuous case are Crocker, McGriffin, Naylor, and Vosper [3]; Scanlon [2]; and Fisk, Caskey, and West [4]. Each of these papers describes a different variation of a force directed placement procedure. Each of these techniques suffers from some technical obstacle which thus limits its effectiveness as a truly useful placement algorithm. It is the objective of this paper to present a refined technique which eliminates these obstacles.

The placement procedure presented here is broken down into two phases, Phase I being the "relative location phase," and Phase II being the "slot assignment or component overlap resolution phase."

II. The Relative Location Problem

We shall define the *relative location problem* as being the task of determining the correct relative position of every component with respect to every other component such that, with a minimum distortion of these positions by the procedures of Phase II, the signal nets of these components can be interconnected using a near minimal amount of wire.

To solve the relative location problem we shall develop a set of force equations governed by the interconnection topology (signal nets) of the system of components, the solution of which produces the desired result. The equations will be based upon the concept of mass as well as the forces of attraction and repulsion between the components. These equations will be formulated with the following characteristics in mind:

1) to behave similarly to the quadratic assignment problem by taking into account the positions of all the components;
2) to tend to minimize the minimum spanning tree (MST) [1] length needed to interconnect the signal nets;
3) to take into account the positions of fixed components such as connectors;
4) to be solvable in an efficient manner.

It would be ideal to have a fifth characteristic which states that "the positions of the slots into which components must be placed would be taken into account;" however, this is impractical from a computational standpoint, since it is analogous to the quadratic assignment problem which requires a solution time that is factorially bounded.

In developing the solution being presented in this paper, numerous concepts were evaluated. Some of these concepts worked well, others did not. A thorough presentation of both the concepts that did and did not work is presented in Quinn [14]. In this paper we will only present those concepts which proved fruitful.

A. Hooke's Law and the Placement of Components

Hooke's Law states that if two particles (components) are connected to each other by a spring (signal net), then there is an attractive force between these two particles (components) that is equal to the spring constant (function of the number of signal names that two components share in common) times the distance between the two particles (components).

Let (x_i,y_i) be the coordinates of component i, and let $\Delta x_{ij}=x_j-x_i$ and $\Delta y_{ij}=y_j-y_i$. Let $K_{ij}=K_{ji}$ be a constant proportional to the number of signals that components i and j have in common;

$$\overline{\Delta S_{ij}}=\begin{pmatrix}\Delta x_{ij}\\ \Delta y_{ij}\end{pmatrix}$$

be the vector distance between components i and j ($\overline{\Delta S_{ij}}=-\overline{\Delta S_{ji}}$), and let $\bar{F}_{ij}$ be the force exerted on component i by component j. Then $\bar{F}_{ij}=-K_{ij}\overline{\Delta S_{ij}}$. The magnitude of $\overline{\Delta S}$ will be denoted by ΔS, and as will be shown later, there are several analytical and computational advantages to choosing as our metric $\Delta S=|\Delta x|+|\Delta y|$ rather than the more conventional Euclidian distance $\Delta S=(\Delta x^2+\Delta y^2)^{1/2}$.

The x and y coordinate components of $\bar{F}_{ij}$, denoted by Fx_{ij} and Fy_{ij}, are given by the expressions

$$Fx_{ij}=-\bar{F}_{ij}\cdot\Delta x_{ij}/\Delta S_{ij}=-K_{ij}\cdot\Delta x_{ij}$$

and

$$Fy_{ij}=-K_{ij}\cdot\Delta y_{ij}.$$

If we assume M moveable components, then the total force on each component in the x and y direction is given by the equations

$$Fx_i=-\sum_{j=1}^{M}K_{ij}\Delta x_{ij}$$

and

$$Fy_i=-\sum_{j=1}^{M}K_{ij}\Delta y_{ij},\qquad \text{for } i=1,2,\cdots,M. \tag{1}$$

A solution to (1) consists in finding a set of values for all the x_i's and y_i's such that all the $Fx_i = Fy_i = 0$. One can quickly see that if all the components are free to move, then the solution to (1) is $x_1 = x_2 = x_3 = \cdots = x_M$ and $y_1 = y_2 = y_3 = \cdots = y_M$, i.e., all components collapse to a single point. This is due to the fact that there are no external forces acting on the system of components to keep them spatially separated.

This set of equations represents the basis for all force directed placement techniques [2]–[4]. The equations give some intelligible results in cases where there are three or more fixed components that have different x and y locations. Crocker *et al.* [3] use this method exclusively, since they are placing components in a package that has four connectors surrounding the four sides of the board. They solve the set of equations algebraically. In most cases, where the board has only one edge connector, this formulation is unacceptable. Hall [15] uses (1) with one additional constraint, i.e., if $\bar{X} = (x_1, x_2, \cdots, x_M)$ is a column vector and $\bar{X}^T$ its transpose, then Hall sets $\bar{X}\bar{X}^T = 1$, and similarly for $\bar{Y}$. This constraint keeps the components spatially separated.

B. Hooke's Law with Repulsion

To keep components spatially separated we have chosen to use a force of repulsion between unconnected components. This type of force is useful since it tends to spread out the components across the board as well as increase the probability of the components taking their correct relative positions.

There are two basic forms that a repulsion term can take, one which is inversely proportional to distance, or one which is constant, i.e., not dependent on distance. Experimental evidence indicates that either type of repulsion gives approximately the same results, but that a solution obtained using a constant repulsion term requires significantly less computer time.

Our formulation of the placement problem, using constant repulsion becomes

$$\bar{F}_{ij} = -K_{ij}\overline{\Delta S_{ij}} + \overline{R_{ij}} = -K_{ij}\begin{pmatrix}\Delta x_{ij}\\ \Delta y_{ij}\end{pmatrix} + \frac{R_{ij}}{\Delta S_{ij}}\begin{pmatrix}\Delta x_{ij}\\ \Delta y_{ij}\end{pmatrix}$$

where

$$R_{ij} = \begin{cases} 0, & \text{for } K_{ij} \neq 0 \\ 0, & \text{for } i = j \\ R, & \text{for } K_{ij} = 0 \end{cases}.$$

$\overline{R_{ij}}$ is a vector whose direction is that of $\overline{\Delta S_{ij}}$ and whose magnitude is R_{ij}. R is defined by the equation

$$R = \left(\frac{1}{C_R}\right)\cdot\left[\sum_{i=1}^{M}\sum_{j=1}^{M} K_{ij}\right]/T$$

where T equals the number of K_{ij}'s equal to 0. An explanation of R will be given shortly.

$\bar{F}_{ij}$ can be broken down into the components

$$Fx_{ij} = F_{ij}\cdot\Delta x_{ij}/\Delta S_{ij} = -K_{ij}\Delta x_{ij} + R_{ij}\Delta x_{ij}/\Delta S_{ij}$$

and

$$Fy_{ij} = F_{ij}\cdot\Delta y_{ij}/\Delta S_{ij} = -K_{ij}\Delta y_{ij} + R_{ij}\Delta y_{ij}/\Delta S_{ij}. \quad (2a)$$

Notice that the repulsion terms for Fx_{ij} and Fy_{ij} contain a scale factor equal to the cosine of the angle that the line connecting the two components makes with the x and y axes, respectively.

Expanding this formulation to cover all M moveable components, we obtain

$$Fx_i = \sum_{j=1}^{M} Fx_{ij} \quad \text{and} \quad Fy_i = \sum_{j=1}^{M} Fy_{ij}. \quad (2b)$$

These two repulsion equations contain the term ΔS_{ij}. This has a slightly undesirable side effect in that it mathematically couples these equations. This coupling effect makes the solution of these equations more difficult because there are now twice as many simultaneous equations.

C. Distance Measure

For both analytic and computational reasons we have chosen to define ΔS_{ij} by the Manhattan distance $|\Delta x_{ij}| + |\Delta y_{ij}|$. Computationally, it is preferred because ΔS_{ij} must be evaluated $M^2/2$ times for each iteration of our algorithm in order to determine the distance from every component to every other component. If the function $(\Delta x_{ij}^2 + \Delta y_{ij}^2)^{1/2}$ were used it would add significantly to the computational time because of the square root function. Analytically it is preferred because the sum of the contribution in the x and y directions equals 1, i.e.,

$$\Delta S = |\Delta x/(|\Delta x| + |\Delta y|)| + |\Delta y/(|\Delta x| + |\Delta y|)| = 1$$

for all values of Δx and Δy. If the Euclidian distance measure is used, then

$$\Delta S = |\Delta x/(\Delta x^2 + \Delta y^2)^{1/2}| + |\Delta y/(\Delta x^2 + \Delta y^2)^{1/2}|.$$

Now $1 \leqslant \Delta S \leqslant \sqrt{2}$, where $\Delta S = 1$ for Δx or $\Delta y = 0$, and $\Delta S = \sqrt{2}$ for $\Delta x = \Delta y \neq 0$. For this measure we get $\sqrt{2}$ greater repulsion along the diagonals than along the axes. This effect is undesirable, and, therefore, we use $\Delta S_{ij} = |\Delta x_{ij}| + |\Delta y_{ij}|$ throughout the remainder of this work. Other researchers [2], [4] have consistently used the Euclidian distance measure.

D. Other Types of Components

There are basically three types of components, namely moveable, fixed, and semimoveable. So far we have limited our discussion to the moveable type components. We will now turn our attention to the other types of components.

Fixed Components: Fixed components, by definition, are rendered immoveable throughout the placement process. These components are kept fixed by setting their positions (x_i, y_i) equal to a constant throughout the placement process. The way this is accomplished in the set of equations is by a simple indexing change on the summation limit. If there are N components total (both fixed and moveable), and M moveable components, then our equations become

$$Fx_i = \sum_{j=1}^{N} -K_{ij}\Delta x_{ij} + R_{ij}\Delta x_{ij}/\Delta S_{ij} \quad (3)$$

$$Fy_i = \sum_{j=1}^{N} -K_{ij}\Delta y_{ij} + R_{ij}\Delta y_{ij}/\Delta S_{ij} \quad (4)$$

for all $i = 1,2,\cdots,M$ moveable components.

The fixed components are generally groups of connector pins which have been preassigned signal names. However, there are many situations when the technology allows for the assignment of the signal names to the connector pins after the placement process. In these cases it is advisable to treat all the groups of connector pins along one edge of the board as a "bar" type connector. The bar type connector attracts components in one direction only. For example, if there is a "bar" type connector on the bottom edge of the board, all components connected to it will be attracted in the downward direction only. To implement this type of connector we need only make a small indexing change to the upper limit of the summation in one dimension.

If there are N components total, M moveable components, and B bar type components (assume the bar components exert a force in the y direction only) then our equations become

$$Fx_i = \sum_{j=1}^{N-B} -K_{ij}\Delta x_{ij} + R_{ij}\Delta x_{ij}/\Delta S_{ij}$$

and

$$Fy_i = \sum_{j=1}^{N} -K_{ij}\Delta y_{ij} + R_{ij}\Delta x_{ij}/\Delta S_{ij} \quad (5)$$

for $i = 1,2,\cdots,M$.

Semimoveable Components: A semimoveable component is free to move along one axis only. This constraint is easily implemented by keeping the component's position in the restricted direction constant. This again requires only a summation limit change in one dimension. If there are N components total, M moveable components, L semimoveable components (assume semimoveable components are free to move in the x direction only) then our equations become

$$Fx_k = \sum_{j=1}^{N} -K_{kj}\Delta x_{kj} + R_{kj}\Delta x_{kj}/\Delta S_{kj}$$

$$Fy_i = \sum_{j=1}^{N} -K_{ij}\Delta y_{ij} + R_{ij}\Delta y_{ij}/\Delta S_{ij} \quad (6)$$

for $i = 1,2,\cdots,M$ and $k = 1,2,\cdots,M+L$, where components $k+1, k+2,\cdots,M+L$ are restricted to move in the x direction only.

E. Effects on Solution by the Fixed and Semimoveable Components

If one solves the system of equations presented so far they would find that it leads to unacceptable results because in some cases the moveable components tend to surround the connectors. This has an adverse effect on the results of Phase II, because the components are not in their correct relative positions for assignment to slots on the board. To help alleviate this problem, we have found it useful to keep the bulk of the components at some fixed location on the board. To accomplish this we recall that to maintain the center of mass of a system of particles at a fixed location it is required to negate the net effect on any external forces acting on the system. In our case, the system of particles is the set of moveable components, and the external forces are those forces exerted on the moveable components by the fixed components. We can compute this external force, which we call the force on the

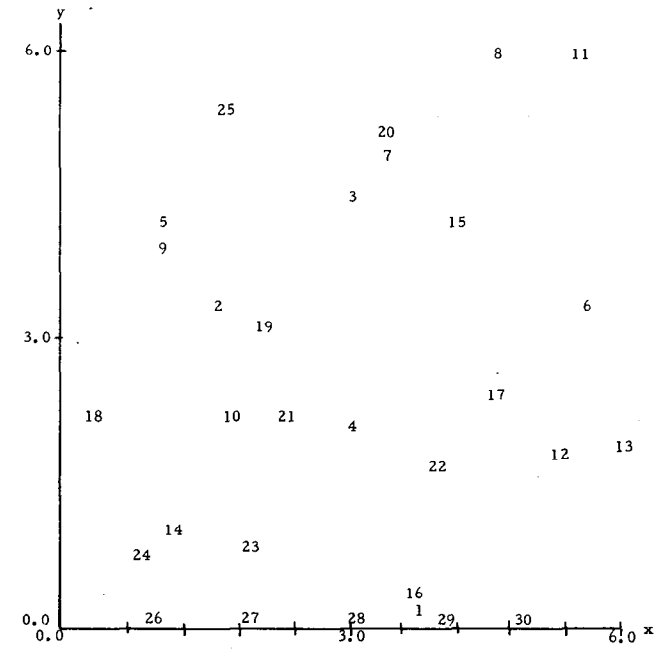


Fig. 1. (IL25) random initial placement of moveable components.

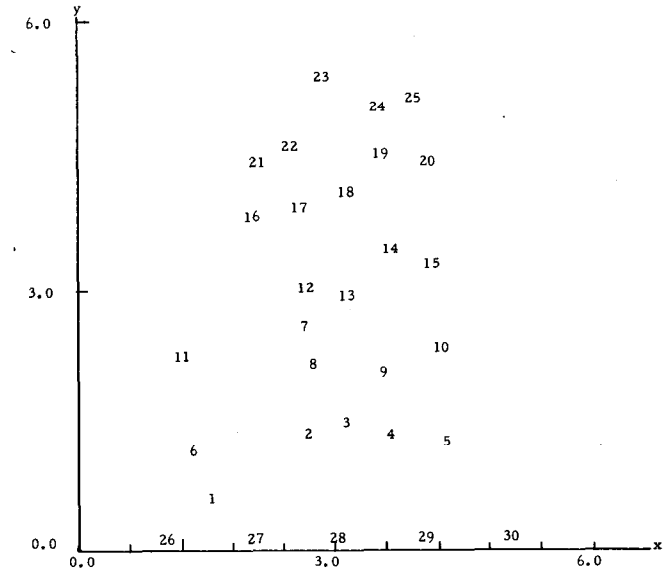


Fig. 2. (IL25) result of Phase I using (8).

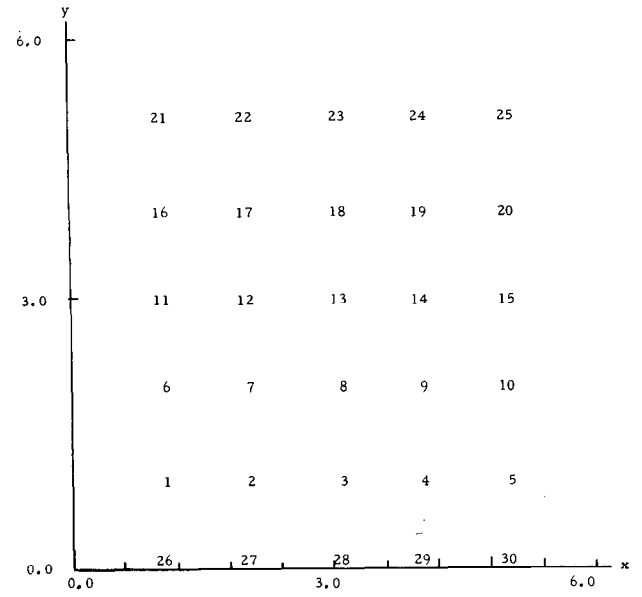


Fig. 3. (IL25) result from Phase II (optimum configuration for components on a 5×6 array board).

center of mass, denoted by *FCMX* and *FCMY*, by the following equations:

$$FCMX = \sum_{i=1}^{M} \sum_{j=M+1}^{N} -K_{ij}\Delta x_{ij}$$

$$FCMY = \sum_{i=1}^{M} \sum_{j=M+1}^{N} -K_{ij}\Delta y_{ij}. \tag{7}$$

Note, however, that

$$FCMX = \sum_{i=1}^{M} Fx_i \quad \text{and} \quad FCMY = \sum_{i=1}^{M} Fy_i$$

where Fx_i and Fy_i are defined in (2b) because the sum of all the forces acting on the moveable components must equal the total external force acting on these components, i.e., the total force between the moveable components and the fixed components.

In order to hold the center of mass (*CM*) of the moveable components fixed we need only to subtract a portion of *FCMX* and *FCMY* from each moveable component. Since there are M moveable components, we subtract from each component $FCMX/M$ and $FCMY/M$, and our final set of force equations becomes

$$Fx_i = \left[\sum_{j=1}^{N} -K_{ij}\Delta x_{ij} + R_{ij}\Delta x_{ij}/\Delta S_{ij}\right] - FCMX/M$$

$$Fy_i = \left[\sum_{j=1}^{N} -K_{ij}\Delta y_{ij} + R_{ij}\Delta y_{ij}/\Delta S_{ij}\right] - FCMY/M \tag{8}$$

for all $i = 1,2,\cdots,M$.

The initial position of the *CM* has a pronounced effect on the results. Its effect is covered in Section V. To the best of our knowledge, this problem of correctly aligning the moveable components with respect to the fixed components has not been previously addressed.

We shall illustrate our technique for obtaining a placement on our test board IL25.[1] This problem was designed so that the optimal placement is known. Components 1–25 are moveable and 26–30 are fixed. This problem had 65 signal nets and 122 edges in the minimal spanning tree. Figs. 1–3 show, respectively, the initial position of the components, the results of Phase I (using (8)), and the result of Phase II positioning upon a 5×6 array board using the results of Phase 1. This final solution is an optimal one.

III. Solution of Force Equations

In this section we will discuss an efficient method for solving the set of equations derived in Section II. Our goal is to develop a procedure that produces a solution at a reasonable cost (CPU time and storage), gives results that are in general comparable to or superior to those of other well-known techniques (Pairwise Interchange, Steinberg–Rutman, Force Directed Pairwise Relaxation, etc.), and obtains results for a variety of types of carriers. We now present our solution technique.

A. Solution Technique

We know *a priori* that the desired solution has the property that the forces on all components are equal to zero. By setting each of the forces in (8) equal to zero, we obtain the following set of algebraic equations:

$$Fx_1 = \left[\sum_{j=1}^{N} -K_{1j}\Delta x_{1j} + R_{1j}\Delta x_{1j}/\Delta S_{1j}\right] - FCMX/M = 0$$

$$\vdots \qquad\qquad \vdots$$

$$Fx_M = \left[\sum_{j=1}^{N} -K_{Mj}\Delta x_{Mj} + R_{Mj}\Delta x_{Mj}/\Delta S_{Mj}\right] - FCMX/M = 0$$

$$Fy_1 = \left[\sum_{j=1}^{N} -K_{1j}\Delta y_{1j} + R_{1j}\Delta y_{1j}/\Delta S_{1j}\right] - FCMY/M = 0$$

$$\vdots \qquad\qquad \vdots$$

$$Fy_M = \left[\sum_{j=1}^{N} -K_{Mj}\Delta y_{Mj} + R_{Mj}\Delta y_{Mj}/\Delta S_{Mj}\right] - FCMY/M = 0. \tag{9}$$

Since these equations are nonlinear, their solution is somewhat difficult to obtain. We have employed a modified version of the Newton–Raphson (NR) [6] technique for solving this system of equations. The NR method is very costly to use in terms of storage requirements and CPU time per iteration. In general, for the $2M$ simultaneous equations in $2M$ unknowns, we must calculate the Jacobian ($4M^2$ partial derivatives), and must then solve for the $2M$ increments p_i, q_i ($i = 1,2,\cdots,M$) by solving a set of $2M$ simultaneous linear equations in $2M$ unknowns. We shall present an easier method which, while perhaps not converging in as few a number of iterations as the NR method, is much easier to apply, and requires less com-

[1] A test board labeled ILn has n components.

putation time. Our method is a slightly modified version of the modified Newton–Raphson method for simultaneous equations (MNR) [6].

The MNR method consists of applying the one-dimensional NR method $2M$ times, once for each variable. Each time it is applied to a variable, the other variables remain fixed at their most current value.

Using this technique the computation time per iteration grows approximately as M^2, since for each iteration we must evaluate each function and its derivative using the newly updated value of the variables. If we make a slight modification to the MNR method by not using the updated values of the variables during an iteration, but instead update the values of the variables at the end of each iteration, then our computation time per iteration grows approximately as $M^2/2$. This reduction is significant for large M. The reason for the reduction in computation is that we can take into account certain geometrical symmetries that exist between the components if we know that they remain fixed during an iteration. The geometrical symmetries exist because the distances between components remains the same throughout each iteration.

Another reason for updating the values of the variables at the end of an iteration is that the *FCMX* and *FCMY* forces can be calculated quite easily if the components remain fixed during an iteration.

Applying this modified version of the NR method to (8), we obtain, for $i=1,2,\cdots,M$,

$$\frac{\partial Fx_i}{\partial x_i} = F'x_i = \sum_{j=1}^{N} -K_{ij} + R_{ij}\frac{|\Delta y_{ij}|}{\Delta S_{ij}^2} + \frac{1}{M}\sum_{j=M+1}^{N} K_{ij}$$

and

$$\frac{\partial Fy_i}{\partial y_i} = F'y_i = \sum_{j=1}^{N} -K_{ij} + R_{ij}\frac{|\Delta x_{ij}|}{\Delta S_{ij}^2} + \frac{1}{M}\sum_{j=M+1}^{N} K_{ij}$$

$$x_i(\text{new}) = x_i - \tfrac{1}{2}Fx_i/F'x_i$$

$$y_i(\text{new}) = y_i - \tfrac{1}{2}Fy_i/F'y_i. \qquad (10)$$

The factor of $1/2$ is due to the fact that any two components are attracted to each other by the same force, therefore we need move each component only $1/2$ the designated distance.

Our solution is derived as follows.

Solution Procedure: Method for solving the force equations.

Step 0: Preliminaries: for $i=1,2,\cdots,N$, set x_i and y_i to their initial values.

Step 1: For $i=1,2,\cdots,M$, set

$$Fx_i = \sum_{j=1}^{N} -K_{ij}\Delta x_{ij} + R_{ij}\Delta x_{ij}/\Delta S_{ij}$$

and

$$Fy_i = \sum_{j=1}^{N} -K_{ij}\Delta y_{ij} + R_{ij}\Delta y_{ij}/\Delta S_{ij}.$$

Step 2: For $i=1,2,\cdots,M$, set

$$F'x_i = \sum_{j=1}^{N} -K_{ij} + R_{ij}|\Delta y_{ij}|/\Delta S_{ij}^2 + \frac{1}{M}\sum_{j=M+1}^{N} K_{ij}$$

and

$$F'y_i = \sum_{j=1}^{N} -K_{ij} + R_{ij}|\Delta x_{ij}|/\Delta S_{ij}^2 + \frac{1}{M}\sum_{j=M+1}^{N} K_{ij}.$$

Step 3: Set

$$FCMX = \sum_{i=1}^{M} Fx_i \quad \text{and} \quad FCMY = \sum_{i=1}^{M} Fy_i.$$

Step 4: For $i=1,2,\cdots,M$, set

$$Fx_i = Fx_i - FCMX/M \quad \text{and} \quad Fy_i = Fy_i - FCMY/M.$$

Step 5: For $i=1,2,\cdots,M$, set

$$x_i = x_i - 1/2\cdot Fx_i/F'x_i \quad \text{and} \quad y_i = y_i - 1/2\cdot Fy_i/F'y_i.$$

Step 6: Set

$$\text{NORM} = \sum_{i=1}^{M} [|Fx_i| + |Fy_i|].$$

IF (NORM $>\epsilon$) GO TO Step 1.

Step 7: PRINT RESULTS.

B. Storage Requirements

This method requires $N(N-1)/2$ cells of storage for the $K=[K_{ij}]$ and $R=[R_{ij}]$ matrices, and $8M$ cells to store the vectors $\overline{Fx}$, $\overline{Fy}$, $\overline{F'x}$, $\overline{F'y}$, $\overline{Fx^{old}}$, $\overline{Fy^{old}}$, $\bar{x}$, $\bar{y}$.

C. Computational Complexity

We will describe the computational complexity per iteration of this method in terms of the number of arithmetic operations required. To do this we will take into account various symmetries that exist, and make some assumptions about the density of the K matrix. The reason for taking into account the symmetries of the problem is self explanatory. The reason for needing to assume the density of the K matrix is that the K and R matrices are mutually exclusive; therefore, the terms involving R need only be evaluated when the K elements are zero, and vice versa.

Table I gives the number of arithmetic operations per step for our solution procedure. For this table we will assume a 10-percent dense K matrix. A CDC Cyber 172 computer was used for all problems in this work and all times are for this machine.

If we assume that a memory reference takes 13 units of time, additions/subtractions take 8 units of time, and multiplies/divides take 32 units of time, then the number of units of time in our procedure is proportional to $24.5MN+217.9M^2+199.6M+99$. (A unit of time on a CDC Cyber 172 is 100 ns.) The number of iterations required to obtain a solution is dependent upon problem size (N), the amount of repulsion (R), and the stopping criterion value (ϵ).

TABLE I
NUMBER OF ARITHMETIC OPERATIONS PER STEP OF SOLUTION PROCEDURE

Step Number	Number Additions/Subtracts	Number Multiply/Divides	Number Memory Reference
1	$.1(4MN-\frac{M}{2}(M+7))$ $+.9(4M(M-1))$	$.1(3MN-\frac{3M}{2}(M+1))$ $+.9(\frac{3}{2}M(M-1))$	$.1(9MN-\frac{M}{2}(5M+13))$ $+.9(11/2M(M-1))$
2	$.9(2M(M-1))$	$.9(M(M-1))$	$.9(4(M(M-1))$
3	2M	0	2M+2
4	2M	0	4M+2
5	4M	4M	10M+1
6	2M+1	0	2M+2

D. Convergence

The question might arise, "does our method guarantee convergence of the force equations?" The answer is yes, because the Fx_i's and Fy_i's represent the partial derivatives of the potential energy function of the system of components, and since this system is dissipative in energy it implies that these derivatives are always pointing in the direction of lower potential energy. Our method always moves the solutions in the direction of these derivatives, hence leading to a convergent solution.

As an aside, we would like to mention that two other techniques for solving (8) were studied in depth. One used numerical integration, where a damping term was used to force a steady state solution. The second was an algebraic technique using the classical NR method. Neither of these techniques was as efficient as the one presented here. In Quinn [16] a solution technique using Fletcher–Reeves method of conjugate gradient minimization was presented. The method presented here produces a solution which is 2 to 3 orders of magnitude more efficient in terms of computer time. It has been called to our attention by one of the referees that the use of a procedure by Brown [17], which is a new variation on Newton's method, may lead to still more efficient computational results.

IV. ASSIGNMENT OF COMPONENTS TO "REAL" LOCATIONS (PHASE II)

The objective in solving the force equations was to find the correct relative position of every component with respect to every other component. In obtaining this solution we often find that the resultant placement is physically unacceptable from either the standpoint that the components overlap, or that the technology requires the components to reside in predetermined locations (slots). In an effort to resolve this apparent conflict we have broken the problem down into two subproblems, which are solved as follows: 1) if we are dealing with array boards, where components must go into slots, then an assignment technique is used; 2) if components are free to reside anywhere on the board, then a technique to resolve component overlap is used. For this latter case, the final placement is done manually, usually via an interactive graphic system.

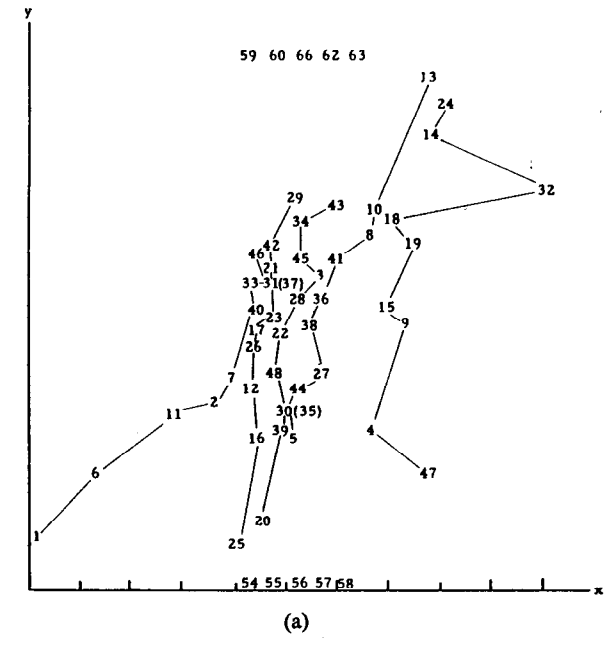
A. Array Carrier

An array carrier is defined as a media having some number Q $(Q \geq M)$ of predefined locations (slots) in which the M moveable components are to be placed. Generally, these slots are in a regular array (5×5, 5×10, 11×15, etc.), with distances between all columns being equidistant and distances between all rows being equidistant. This type of carrier is most often encountered when dealing with integrated circuit packages to be placed on a printed circuit board, in the design of master slice logic circuits, or in PLA's.

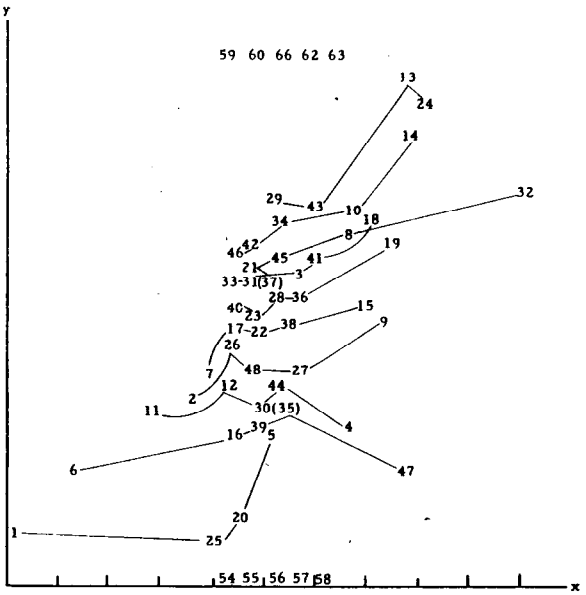
Since the results of Phase I give us the relative positioning of the components, we would like to distort this positioning as little as possible when assigning components to slots. That is, we would like to assign components to slots in such a way so as to minimize the total distortion of the placement. To accomplish this we have employed the linear assignment algorithm [8]. We have chosen to use Bourgeois and Lassalle's [11] implementation of Munkres' [8] formulation of the Hungarian Assignment algorithm in solving this problem. The assignment algorithm requires the generation of a cost matrix, where each element C_{ij} of the cost matrix represents the cost of assigning component j to slot i. We considered three cost functions, namely ones based upon rectilinear distance from the component to slot, the Euclidian distance and the square of the Euclidian distance.

Using rectilinear or Euclidian distance we can show analytically that the relative position of the components obtained in Phase I is not preserved. Therefore, we have chosen to use the square of the Euclidian distances. One reason for the success of the cost function can be seen from the mathematics involved. Note that even though we are physically moving components in two dimensions, when we use the assignment algorithm we are really working in M-dimensional space. That is, the linear assignment algorithm for assigning M components to Q slots $(M \leq Q)$ dictates that we find a permutation $P = \{p_1, p_2, \cdots, p_M\}$ on the integers $1, 2, \cdots, Q$ taken M at a time such that $\sum_{i=1}^{M} C_{ip_i}$ is minimum over all such permutations. This sum represents the magnitude of an M-dimensional cost vector, and the distance of any M-dimensional vector in Euclidian space is equal to the square root of the sum of the squares of all its components. Therefore, since we are endeavoring to find the minimum of the total distance traveled by all components, we must compute the square root of the sum of the squares of all the distances traveled. Using this measure gives us the desired result for minimizing the total distortion of the components. It should be pointed out that others [2] who have attempted to use this type of technique have failed to realize the necessity of using this cost function. Hence, the quality of their results has been correspondingly reduced.

In Fig. 4 we show a placement obtained from Phase I, and in Fig. 5 the final placement obtained from Phase II. In Fig. 4(a) ((b)) we have drawn lines connecting those



(a)



(b)

Fig. 4. (IC66) placement after Phase I. (a) Lines indicating columns from Phase II. (b) Lines indicating rows from Phase II.

components that are assigned to the same column (row) in Fig. 5. When rectangular or Euclidian distance measures are used, there are usually numerous intersections of lines implying that relative positions are not kept. This is not true when the square of the Euclidian distance is used.

Starting with the results shown in Fig. 4, the MST distances for the results of Phase II using rectilinear, Euclidian, and Euclidian distance squared are 963, 920, and 907 units, respectively.

Analysis and Comments on the Use of the Assignment Algorithm: One should note that Phase II of our process is conceptually quite different than Phase I. In Phase II, the interconnection topology of the components is no longer considered; only the physical positioning of the

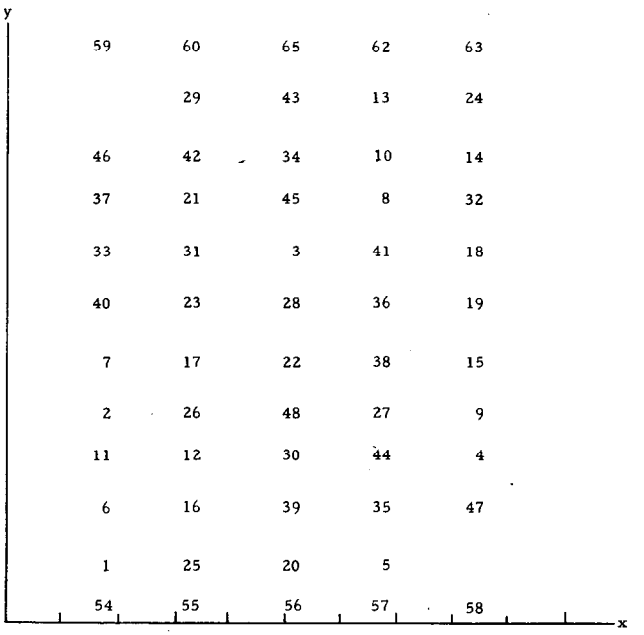


Fig. 5. (IC66) final placement after Phase II using Euclidian distance squared for assignment cost.

components is taken into account. This disregard for the interconnection topology can have some deleterious side effects. For example, if the results of Phase I dictate that the optimum board configuration is a 5×5 arrangement of slots, and our board consists of a 2×13 arrangement of slots, then the results of Phase II may not give the solution with the shortest total wire length. We have studied this situation for the case of a 5×5 lattice structure and have found that the difference in wire length between our 2×13 solution and the optimum 2×13 solution is less than 1 percent.

The computational complexity of the assignment algorithm is proportional to M^2Q. This implies that Phase II of the placement process requires more computation than Phase I, which is proportional to M^2.

B. Nonarray Carriers

Nonarray carriers are generally media containing a variety of different types and sizes of components. For example, an "analog" PCB usually contains IC's, transistors, capacitors, resistors, etc. In the past the layout of these types of carriers has traditionally been done manually. Components usually need not be aligned into columns and rows.

In the case where the assignment of components to slots is not a requirement, we must spread the components out over the board, eliminating any overlap that may result from Phase I. The elimination of overlap is done automatically by our Phase II program. It is carried out in such a way so as to preserve, as much as possible, the relative positioning of the components.

The implementation of our Phase I and II placement techniques has been embedded into an interactive graphics packaging program. This system allows the user to

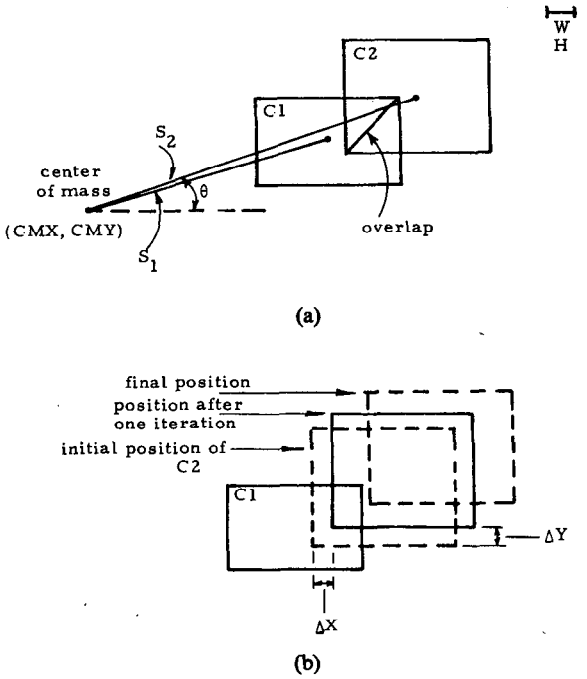


Fig. 6. Example of component overlap resolution.

view, via a vector graphics display, the positions of all components as found via our Phase I and II algorithms. It also allows the user to move any particular component to any position on the board by dragging it with a light pen. While moving the component, the program displays, in a "rubber band" fashion, the various places to which the component is connected. In this manner the user can easily find a good final position for all components. We have found this approach to be highly successful.

C. Resolution of Component Overlap

The resolution of component overlap is done on a pairwise basis. For M components, all $M(M-1)/2$ pairs of components are compared. If two components overlap, we move the component furthest away from the center of mass of the board, say component C, radially outward by some computed amount $(\Delta X_C, \Delta Y_C)$.

The set of pairs is processed iteratively. The process terminates when a sequence of $M(M-1)/2$ pairs of components are processed and no overlap exists.

In moving a component its overlap with another component may not be completely eliminated. In addition, new overlaps may be created when a component is moved. In any event, the process does readily converge and is effective in keeping the relative position of components in the final placement quite similar to the relative position as found by our Phase I algorithm.

In Fig. 6(a) we illustrate the initial location of two components $C1$ and $C2$, as well as the quantities S_1, S_2, θ and overlap. In Fig. 6(b) we show the result from one iteration of our procedure, as well as the final position obtained after several iterations.

Let W be one-half the width of the narrowest component, and H be one-half the height of the shortest component. Then the distance from the center of mass to the center of component Ci is $S_i = ((X_{Ci} - CMX)^2 + (Y_{Ci} - CMY)^2)^{1/2}$.

From Fig. 6(a), we see that $S_2 > S_1$, hence $C2$ is moved radially outward away from the center of mass. The quantities ΔX_{C2} and ΔY_{C2} are computed according to the equations

$$\Delta X_{C2} = \sigma_1 \cdot \min(\text{overlap}, W)$$

and

$$\Delta Y_{C2} = \sigma_2 \cdot \min(\text{overlap}, H)$$

where

$$\sigma_1 = \cos\theta = (X_{C2} - CMX)/S_2$$

and

$$\sigma_2 = \sin\theta = (Y_{C2} - CMY)/S_2.$$

The amount a component is moved in the x and y direction is restricted to be less than W and H, respectively, so that large components do not jump over smaller components. The sine and cosine functions are used so that components move out radially, i.e., in the direction of a vector from the center of mass to $C2$.

V. Experimental Results

In this section we will briefly review some of the results obtained by using this system. Throughout this chapter we will use the sum of the minimal spanning tree for all nets as a measure of the "quality" of a placement. For uniformity, the distance between each row and column in an array of slots will be taken as one unit.

A. Parameteric Analysis

The following list indicates several parameters of our procedure which we have studied:

1) initial placement of components,
2) problem size (number of components),
3) position of the center of mass (CM) of the moveable components,
4) repulsion constant,
5) interconnection matrix (how it represents the net list).

A complete analysis of our results can be found in Quinn [14]. In this section we will present a brief summary of the results obtained.

We used 12 test boards in evaluating our system. A few were "theoretical boards," constructed in such a way so that the optimal placement was known. Seven were real industrial boards. Boards labeled IC67, 108, 116, 136, and 151 are real boards taken from Stevens' [13] dissertation. Boards IC 67 and 151 were also processed by Hanan [12].

Initial Placement: It has been shown [12] that iterative improvement techniques tend to work better if preceded by some sort of constructive initial placement procedure (CIPP). This is also the case with our method. However, our CIPP is considerably different from the normally

recognized techniques. It consists of solving the force equations (8) using no repulsion ($R_{ij}=0\ \forall i,j$). The only forces keeping the system of components from totally collapsing are the center of mass forces. This CIPP is preceded by randomly placing the moveable components uniformly over the board on a continuous basis. By eliminating the R_{ij} term from (8), the resulting equations become linear and can be solved using classical techniques for the solution of simultaneous linear equations. Rather than develop this additional code, we chose to use our existing code to solve these equations while setting $R_{ij}=0$. This CIPP typically takes only a few (20–50) iterations to converge and takes considerably less computation per iteration than when repulsion is used. The CIPP phase takes about 2–8 percent of the total time required to find a placement.

Our experimental results indicate that the interconnection lengths obtained with and without using a constructive initial placement are approximately equal. This points out the relative insensitivity of this procedure to the initial positioning of the components. The most significant observation is that in almost all cases the procedure finds the same solution when our CIPP is used, independent of the initial random placement. The other very important observation, and in fact the reason for using CIPP, is that in most cases the total computer time required to obtain a solution is significantly reduced.

Problem Size: Problem size primarily affects the computational time for each iteration. Also the number of iterations for convergence appears to be more interconnection topology dependent than problem size dependent.

Position of the Center of Mass: The position of the *CM* in the vertical direction can have a pronounced effect upon the placement results. To see this, assume the connector is at the bottom of the board and recall that the position of the *CM* is at the geometrical center of the carrier to which components are to be assigned by Phase II. Note that all the components are held on the carrier away from the connectors. This positioning may distort the actual relative positioning in which the components would like to be arranged because the *FCMY* forces are excessively large. The large *FCMY* forces push those components that are weakly connected to the rest of the components further up on the carrier than required. The solution to this problem is to force the *CM* to a position that allows the components to end up as close to the connectors as possible without going below them. The determination of this position is possible by dynamically allowing the *CM* to move downward until the y position of the lowest moveable component goes to zero.

Phase I Stopping Criteria: The relative location process of Phase I is theoretically stopped when all the forces on the components are zero. This criteria is impractical from a computational standpoint. Therefore, we stop the process when

$$\sum_{j=1}^{N} [\,|Fx_i|+|Fy_i|\,] \leqslant \epsilon.$$

TABLE II
SOLUTION VERSUS-STOPPING CRITERIA

MST of \ ϵ	.001	.01	.1	1.0	10.0	100
IC136	2749	2762	2746	2738	2740	2763
IC151	2025	2025	2025	2034	2052	2053
IC67	713	713	713	719	764	775
2IC66	907	907	910	910	920	949
IL100	494	494	494	494	516	515
Number of Iterations						
IC136	456	366	279	193	49	14
IC151	388	319	239	162	93	35
IC67	184	151	120	90	52	5
2IC66	146	124	105	89	36	5
IL100	386	291	196	108	51	18

Our studies indicated that the number of iterations varied as a function of $(\log \epsilon)^{-1}$. Usually $\epsilon=1.0$ produced the same results as those obtained using much smaller values of ϵ. Also for $\epsilon>10$ the final results began diverging from the optimal. Some results are shown in Table II.

Repulsion Constant: The repulsive force between unconnected components is the single most critical parameter of the system. It affects the number of iterations as well as the quality of the results.

Evolution of the Equation for R: After much analysis and experimentation, it became evident that R had to be a function of M as well as the interconnection topology. For this reason R was made equal to $(1/C_R)\cdot(\Sigma_{i=1}^{N}\Sigma_{j=1}^{N}|K_{ij}|)/T$, (where T is the number of $K_{ij}=0$ terms), that is, R is proportional to the total attractive force and inversely proportional to the number of times R is used. This value for R has shown a great deal of insensitivity to different types of problems.

Determination of C_R: The method with which we chose to determine C_R was to make a series of runs on different problems, plotting C_R versus a number of normalized quantities such as 1) number of iterations for Phase I convergence, 2) MST of the result of Phase II with and without connector connections, and 3) run time of Phase I and Phase II combined. It is evident from our results that the optimum value of C_R is in the range $1\leqslant C_R\leqslant 2$. It is still an open question as to exactly how to choose the optimum value of C_R for any given problem.

Interconnection Matrix (K_{ij}'s): The K_{ij}'s form the interconnection matrix and represent how the various components attract and repel each other throughout the relative location process. There are several ways of representing a signal set by the K_{ij}'s: 1) as a complete graph where every node is connected to every other node in the net, 2) as a MST, 3) as a string, where no node is connected to more than 2 other nodes, and 4) as a star, where all sink nodes are only connected to the source node.

All the figures and tables in this section were derived using the complete graph as representing the signal nets. For a net having S nodes, K_{ij} is set to $2/S$, because $2/S$ is that fraction of the $S(S-1)/2$ possible edges that are needed to interconnect the S nodes.

TABLE III
COMPARISON OF OUR RESULTS WITH HANAN'S AND STEVENS'

QUINN-BREUER

Board Number	MST
IC67	710
IC108	1089
IC116	1519
IC136	2558
IC151	1915

STEVENS

Board Number	MST of CIPP	MST of PI	MST of SB
IC67	725	700	702
IC108	1230	1085	1122
IC116	1429	1320	1320
IC136	3306	2845	2733
IC151	2318	2243	2181

HANAN

Board #IC67		Initial Placement	Final Distance				
		Random	PI	NI	PIQ	NIQ	FDPRQ
10 Random Starts	Worst Dist.	1460	730	782	780	825	807
	Avg. Dist.	1394	712	750	740	771	750
	Best Dist.	1328	670	680	702	725	720
		Constructive					
	Dist.	812	691	682	717	726	695

Board #IC151		Initial Placement					
		Random	PI	NI	PIQ	NIQ	FDPRQ
10 Random Starts	Worst Dist.	4515	2122	2210	2240	2436	2255
	Avg. Dist.	4427	2025	2120	2100	2252	2146
	Best Dist.	4425	1890	1920	1997	2118	2026
		Constructive					
	Dist.	2046	2117	2170	2088	2126	2090

We investigated using the MST to represent signal nets. In about 60 percent of the cases there was some improvement, usually quite small, in using the MST representation. The CPU time increases significantly for this mode of operation, since the MST can only be computed after having first computed a solution using the complete graph. We therefore concluded that using the complete graph representation was the most desirable.

B. Comparison with Other Published Results

In [13], Stevens published net lists for five ILLIAC IV boards. He also gave the results for these boards using three different placement techniques, namely, a serial technique, a pairwise interchange (PI) technique, and the Steinberg (SB)–Rutman technique [9], [10]. Hanan [12] has also published results for two of these boards using a number of different methods, namely PI, neighborhood interchange (NI), force directed pairwise relaxation (FDPR), and SB. A description of each of these methods is given in [7]. For each of these methods, he makes a distinction between using the MST and the associated quadratic assignment problem (QAP). The acronym for each method is appended with a Q when QAP is used. Table III gives a comparison of our results with those of Stevens and Hanan. The results of Tables II and III differ because Table II was derived holding all parameter fixed except for ϵ, while Table III was derived using what was considered to be the optimum set of parameters for each board.

The results shown in Table III are self-explanatory. There is purposely no comparison given of computer time because all three sets of results were obtained using computers having considerably different characteristics.

C. Solution of Lattice Type Boards

We studied four boards having a lattice type interconnection structure and for which the optimal placement was known. In each case we obtained the optimal placement using our system. In a private communication with Dr. A. Patel of IBM he indicated that he processed two of these boards using both the classical pairwise interchange method and the SR procedure, and neither procedure produced an optimal result.

VI. CONCLUSIONS

We have presented a new formulation of the concept of force-directed placement, as well as an efficient procedure to solve the resulting mathematical equations. The technique gives excellent results both quantitatively, i.e., in finding the correct solution to known optimal problems, and qualitatively, when comparing with other published results. The procedure is economical to use, costing approximately $32 to place the IC151 board. This is equivalent to about 2 man-hours in cost.

At present, force directed procedures appear to be the most suitable types of procedures to offer the designer the capability of placing components on continuous boards. These procedures are also quite applicable for array type boards. Only in cases where the absolute minimum MST length is required should a method such as PI be used. From the results shown in Table III one can see that PI has the potential for giving better results than ours but at a great increase in cost. The PI results of Stevens [13] and Hanan [12] were obtained by taking the best result from ten random initial starting conditions. The average CPU times for PI on the IC151 board for Stevens' work on a Burroughs 6500 was 4020 s, and for Hanan's work on an IBM 360/91 was 485 s. Our procedure gives uniform results, independent of the initial starting, implying that one need make only one run rather than making multiple runs which is usually required if near optimal results are required when using PI. We estimate that our procedure is comparable in cost to the force directed pairwise relaxation technique of Hanan, and typically gives better results.

REFERENCES

[1] V. Kevin and M. Whitney, "Minimum spanning tree," Algorithm 422, *Comm. ACM*, vol. 15, no. 4, pp. 273–274, Apr. 1977.

[2] F. T. Scanlon, "Automated placement of multi-terminal components," *Proc. Design Automation Workshop*, pp. 143–154, July 1971.

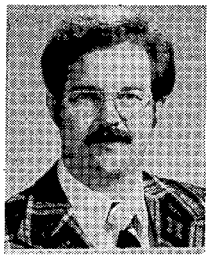
[3] N. R. Crocker *et al.*, "Computer aided placement and routing of high density chip interconnection systems," International Computers Limited, Microsystems Division, Research and Advanced Development Organization Tech. Report, Wenlock Way, West, Gordon, Manchester M12 5DR, England, Aug. 1972.

[4] C. J. Fisk, D. L. Caskey, and L. E. West, "ACCEL: Automated circuit card etching layout," *Proc. IEEE*, vol. 55, pp. 1971–1982, Nov. 1967.

[5] M. Hanan and J. M. Kurtzberg, "Placement techniques," in *Design Automation of Digital Systems: Theory and Techniques*, Ed. M. A. Breuer. New York: Prentice-Hall, 1972, ch. 5.

[6] P. A. Stark, *Introduction to Numerical Methods*. New York: Macmillan, 1970.

[7] *Design Automation of Digital Systems: Theory and Techniques*, Ed. M. A. Breuer. New York: Prentice-Hall, 1972.
[8] J. Munkres, "Algorithms for the assignment and transportation problems," *J. SIAM*, vol. 5, pp. 32–38, 1957.
[9] L. Steinberg, "The backboard wiring problem: A placement algorithm," *SIAM Review*, vol. 3, no. 1, pp. 37–50, Jan. 1961.
[10] R. A. Rutman, "An algorithm for placement of interconnected elements based on minimum wire length," *Proc. 1964 SJCC*, pp. 477–491.
[11] F. Bourgeois and J. C. Lassalle, "An extension of the Munkres algorithm for the assignment problem to rectangular matrices," *Comm. ACM*, vol. 12, pp. 802–806, Dec. 1971.
[12] M. Hanan, P. R. Wolff, and B. J. Agule, "A study of placement techniques for computer logic graphs," *Proc. 13th Design Automation Conf.*, June 1976, pp. 214–224.
[13] J. E. Stevens, "Fast heuristic techniques for placing and wiring printed circuit boards," Ph.D. thesis, Computer Science Dept., University of Illinois, Urbana, 1972.
[14] N. R. Quinn, Jr., "An analysis of a forced directed component placement procedure for printed circuit boards," Ph.D. dissertation, University of Southern California, Jan. 1977.
[15] K. M. Hall, "An r-dimensional quadratic placement algorithm," *Management Sci.*, vol. 17, pp. 219–229, Nov. 1970.
[16] N. R. Quinn, Jr., "The placement problem as viewed from the physics of classical mechanics," *Proc. 12th Design Automation Conf.*, pp. 173–178, June 23–25, 1975.
[17] K. M. Brown, "A quadratically convergent Newton-like method based upon Gaussian elimination," *SIAM J. Numerical Analysis*, vol. 6, pp. 560–569, Dec. 1969.



Neil R. Quinn, Jr., was born in Los Angeles, CA, on February 15, 1945. He received the B.S. degree in physics from Loyola University of Los Angeles, CA, in 1967, and the M.S. degree in computer science and the Ph.D. in electrical engineering from the University of Southern California, Los Angeles, in 1970 and 1977, respectively.

He worked at General Dynamics Corporation in Pomona, CA, from 1967 to 1979. He was involved in the areas of computer aided design and manufacturing and hybrid computing until 1975. In 1976, he was appointed Supervisor of Hybrid Computing Development. In 1976, he was appointed Chief of Engineering Data Systems which involves directing all of the Engineering computing activities for the Pomona Division of General Dynamics. He is presently Vice-President of systems and operations at Environmental Management Systems, Inc., and is working on problems in the area of computer aided energy management. In addition, he is involved in consulting activities in the areas of electronic packaging.

Dr. Quinn is a member of the Association for Computing Machinery.

Melvin A. Breuer (S'58–M'65–SM'73) was born in Los Angeles, CA, on February 1, 1938. He received the B.S. degree in engineering with honors from the University of California, Los Angeles, in 1959, and the M.S. degree in engineering, also from the University of California, Los Angeles, in 1961. In 1965 he received his Ph.D. degree in electrical engineering from the University of California, Berkeley.

In 1965 he joined the staff of the Electrical Engineering Department of the University of Southern California, Los Angeles, where he is currently a Professor. His main interests are in the area of switching theory, computer aided design of computers, fault test generation, and simulation. He is the editor and coauthor of *Design Automation of Digital Systems: Theory and Techniques* (Prentice-Hall); editor of *Digital System Design Automation: Languages, Simulation and Data Base*, Computer Science Press; coauthor of *Diagnosis and Reliable Design of Digital Systems*, (Computer Science Press); and editor-in-chief of the *Journal of Design Automation and Fault Tolerant Computing*. In addition to his research activities, he has been at the forefront of developing courses on design automation and fault tolerant computing at universities and at a number of research institutes.

Dr. Breuer is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.