Reports

Fig. 1. (**a**) A graph with four vertices and six edges. (**b**) A planar graph. (**c** and **d**) Possible planar graphs based on the single-row routing representation.

A Near-Optimum Parallel Planarization Algorithm

YOSHIYASU TAKEFUJI AND KUO-CHUN LEE

A near-optimum parallel planarization algorithm is presented. The planarization algorithm, which is designed to embed a graph on a plane, uses a large number of simple processing elements called neurons. The proposed system, composed of an $N \times N$ neural network array (where N is the number of vertices), not only generates a near-maximal planar subgraph from a nonplanar graph or a planar graph but also embeds the subgraph on a single plane within $0(1)$ time. The algorithm can be used in multiple-layer problems such as designing printed circuit boards and routing very-large-scale integration circuits.

Fig. 2. Violation conditions in the single-row representation (see text).

Table 1. The simulation result: the relation between the coefficients, the number of iteration steps, and the number of embedded edges.

Coefficients		Number of iteration steps	Number of embedded edges
A	B		
2.5	1	6	20
2	1	9	20
1.5	1	7	20
1.25	1	13	20
1	1	9	19
1	1.5	9	19
1	2	9	19
1	2.5	9	19

MAXIMAL PLANARIZATION OF A planar or nonplanar graph is an important problem in designing printed circuit boards and routing very-large-scale integration (VLSI) circuits. A graph is planar if it can be drawn on a single plane with no two edges crossing each other except at their end vertices. If a given circuit is planar, it can be wired on a single layer. If a circuit is nonplanar, it would be desirable to maximize the number of edges to be planarized and minimize the number of edges to be removed from a nonplanar graph. To yield a maximal planar subgraph from a nonplanar graph is an NP-complete problem (NP, nondeterministic polynomial) (1).

Two tasks must be accomplished in solving the planarization problem. One is to verify whether the circuit is planar or not and then to extract a planar subgraph from the nonplanar graph in order to embed it on a plane, called planarity testing. In 1989 Jayakumar et al. proposed a $0(N^2)$ near-optimum planarity testing algorithm (2), where N is the number of vertices. The other task is to embed the planar subgraph on a plane, called plane embedding, so as to maximize the number of edges in the graph. A plane-embedding algorithm was reported by Tarjan in 1971 (3). Unfortunately, few parallel algorithms have been reported to solve the planarization problem.

This paper introduces a parallel planarization algorithm. The algorithm not only yields a near-maximal planar subgraph from a nonplanar graph or a planar graph but also embeds the subgraph on a single layer within $0(1)$ time. In other words, the algorithm provides the necessary and sufficient routing information for embedding a planar sub-

Department of Electrical Engineering, Center For Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, OH 44106.

graph from the given nonplanar or planar graph. In the algorithm the single-row neural representation is introduced. Our simulator found the new maximal planar subgraph in the graph with 10 vertices and 22 edges.

The algorithm uses a massive number of simple processing elements. The processing element is called a neuron, because it performs the function of a simplified biological neuron or a binary neuron (4). The first neural network representation for solving optimization problems was introduced by Hopfield and Tank (5). Szu used the binary neural network for the traveling salesman problem (4). Takefuji proved that the state of the binary neural network is always able to converge to the local minimum (6).

Consider the simple undirected graph composed of four vertices and six edges as shown in Fig. 1a. The graph is planar as long as two edges, (1,3) and (2,4), do not cross each other. Figure 1b shows a planar graph. In the single-row routing representation used here, connection is established by either an upper edge or a lower edge. Two neurons (Vup_{ij} and $Vdown_{ij}$) express the upper and lower line connection between the ith and jth vertices, respectively. For example, the state of two neurons ($Vup_{ij} = 1$, $Vdown_{ij} = 0$) indicates that the edge (i,j) is established by embedding the upper line connection. The following states ($Vup_{ij} = Vdown_{ij} = 0$), ($Vup_{ij} = 0$, $Vdown_{ij} = 1$), and ($Vup_{ij} = Vdown_{ij} = 1$) express no connection, lower line connection, and double-line connection violation, respectively. Figure 1, c and d, shows possible planar graphs based on the single-row routing representation. The number of edges in a given graph determines the number of neurons required. Actually the system requires $2M$ neurons, where M is the number of edges in a graph. For example, the graphs shown in Fig. 1, c and d, can be
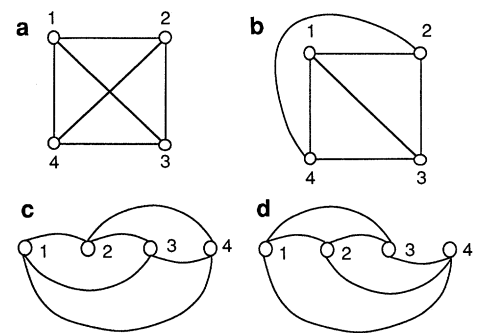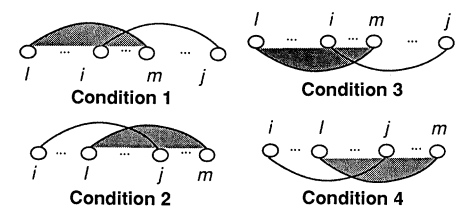
represented by a 2×6 two-dimensional neural network array. The graph in Fig. 1c is expressed by ($Vup_{12} = 1$, $Vdown_{12} = 0$, $Vup_{13} = 0$, $Vdown_{13} = 1$, $Vup_{14} = 0$, $Vdown_{14} = 1$, $Vup_{23} = 1$, $Vdown_{23} = 0$, $Vup_{24} = 1$, $Vdown_{24} = 0$, $Vup_{34} = 0$, $Vdown_{34} = 1$). The graph in Fig. 1d is given by ($Vup_{12} = 1$, $Vdown_{12} = 0$, $Vup_{13} = 1$, $Vdown_{13} = 0$, $Vup_{14} = 0$, $Vdown_{14} = 1$, $Vup_{23} = 1$, $Vdown_{23} = 0$, $Vdown_{24} = 0$, $Vdown_{24} = 1$, $Vup_{34} = 0$, $Vdown_{34} = 1$).

Two kinds of forces perform in the neural network, excitatory and inhibitory forces. In the planarization problem, if an edge (i,j) exists in a given graph, then the ijth neuron to represent embedding the connection line in a plane is encouraged to fire as the excitatory force. The inhibitory force means that the neurons that violate the two-edge-crossing condition are discouraged from firing. The two-edge-crossing violation condition is expressed by the following:

If $V\mathrm{up}_{\ell m} = 1$ and ($\ell < i < m < j$ or $i < \ell < j < m$), then the ijth up-neuron should not be fired. In other words, $V\mathrm{up}_{ij}$ should not be 1.

If $V\mathrm{down}_{\ell m} = 1$ and ($\ell < i < m < j$ or $i < \ell < j < m$), then $V\mathrm{down}_{ij}$ should not be 1. Figure 2 describes the violation conditions for $V\mathrm{up}_{ij}$ and $V\mathrm{down}_{ij}$, which are given by the following functions. For $V\mathrm{up}_{ij}$:

$$\sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m)f(i,m,j)\,V\mathrm{up}_{\ell m}$$

Condition 1

and

$$\sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j)f(\ell,j,m)\,V\mathrm{up}_{\ell m}$$

Condition 2

For $V\mathrm{down}_{ij}$:

$$\sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m)f(i,m,j)\,V\mathrm{down}_{\ell m}$$

Condition 3

and

$$\sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j)f(\ell,j,m)\,V\mathrm{down}_{\ell m}$$

Condition 4

where the function $f(L,M,R)$ is 1 if $L < M < R$, 0 otherwise.

Therefore, the motion equation of the ijth up-neuron is given by:

$$\frac{dU\mathrm{up}_{ij}}{dt} = -A(V\mathrm{up}_{ij} + V\mathrm{down}_{ij} - 1)$$

$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m)f(i,m,j)\,V\mathrm{up}_{\ell m}$$

$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j)f(\ell,j,m)\,V\mathrm{up}_{\ell m} \quad (1)$$

The first term in Eq. 1 performs the excitatory force as long as no two-neuron ($V\mathrm{up}_{ij}$ and $V\mathrm{down}_{ij}$) is fired ($V\mathrm{up}_{ij} = V\mathrm{down}_{ij} = 0$). If ($V\mathrm{up}_{ij} = 1$ and $V\mathrm{down}_{ij} = 0$) or ($V\mathrm{up}_{ij} = 0$ and $V\mathrm{down}_{ij} = 1$), then the first term will not change its output. If ($V\mathrm{up}_{ij} = V\mathrm{down}_{ij} = 1$), then it will act as the inhibitory force. The second term and the last term in Eq. 1 are always inhibitory forces.

The motion equation of the ijth down-neuron is as follows:

$$\frac{dU\mathrm{down}_{ij}}{dt} = -A(V\mathrm{up}_{ij} + V\mathrm{down}_{ij} - 1)$$

$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m)f(i,m,j)\,V\mathrm{down}_{\ell m}$$

$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j)f(\ell,j,m)\,V\mathrm{down}_{\ell m} \quad (2)$$

In general, the planarization problem for a graph with N vertices and M edges can be solved by an $N \times N$ neural network. The

motion equation of the ijth X neuron, where X is either up or down, is:

$$\frac{dUx_{ij}}{dt} = -A(V\mathrm{up}_{ij} + V\mathrm{down}_{ij} - C_{ij})$$

$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m)f(i,m,j)\,Vx_{\ell m}$$

$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j)f(\ell,j,m)\,Vx_{\ell m} \quad (3)$$

where C_{ij} is 1 if the edge (i,j) exists in the given graph, 0 otherwise.

The following procedure describes the proposed algorithm based on the first-order Euler method. It generates a near-maximal
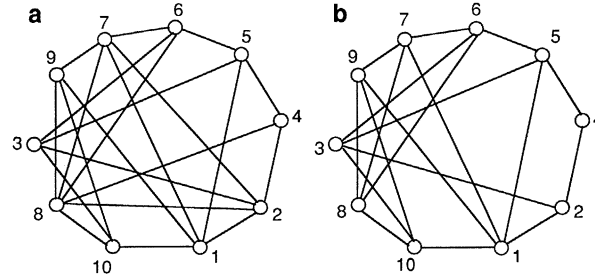

Fig. 3. (a) Nonplanar graph with 10 vertices and 22 edges. [Adapted from (2)] (b) Maximal planar subgraph.
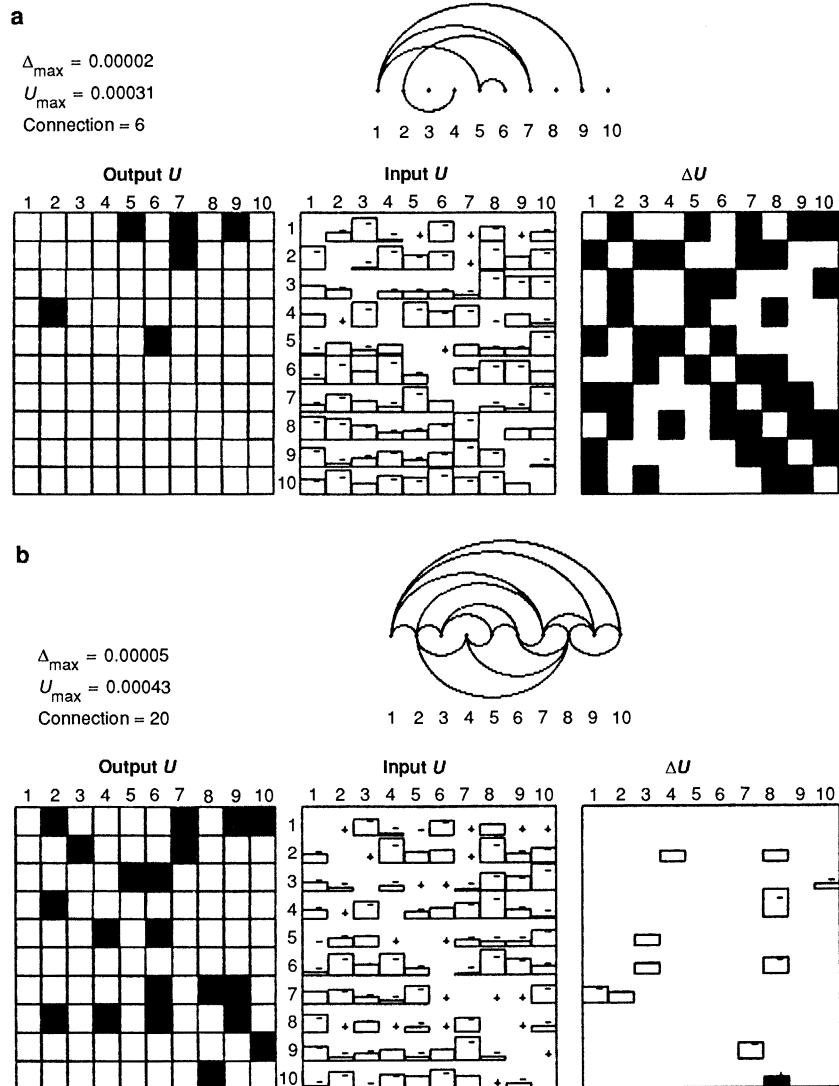

Fig. 4. Convergence of the planarization neural network to a solution. Squares in the upper ($i < j$) and lower ($i > j$) triangle indicate $V\mathrm{up}_{ij}$ and $V\mathrm{down}_{ij}$, respectively. The linear dimension of each rectangle is proportional to the value of $\Delta U\mathrm{up}_{ij}$, $\Delta U\mathrm{down}_{ij}$, $U\mathrm{up}_{ij}$, $U\mathrm{down}_{ij}$, $V\mathrm{up}_{ij}$, and $V\mathrm{down}_{ij}$. Black and white rectangles indicate positive and negative values, respectively. (a) Intermediate state of 44 neurons after the first iteration. (b) Final state of 44 neurons after the 14th iteration. $\Delta_{\max}$ indicates the maximum value in $\Delta U\mathrm{up}_{ij}$ and $\Delta U\mathrm{down}_{ij}$; $U_{\max}$ indicates the maximum value in $U\mathrm{up}_{ij}$ and $U\mathrm{down}_{ij}$. Connection indicates the number of embedded edges.

planar subgraph from the nonplanar or planar graph and embeds it on a plane.

0) Set $t = 0$.

1) Randomize the initial values of $U\mathrm{up}_{ij}(t)$ and $U\mathrm{down}_{ij}(t)$, where $i = 1,...,N$ and $j = 1,...,N$, in the range $-\omega$ to 0; ω is a real number.

2) Evaluate values of $V\mathrm{up}_{ij}(t)$ and $V\mathrm{down}_{ij}(t)$ on the basis of the binary function, where $i = 1,...,N$ and $j = 1,...,N$.

$$V\mathrm{up}_{ij}(t) = f[U\mathrm{up}_{ij}(t)] =$$
$$\begin{cases} 1 & \text{if } U\mathrm{up}_{ij}(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$V\mathrm{down}_{ij}(t) = f[U\mathrm{down}_{ij}(t)] =$$
$$\begin{cases} 1 & \text{if } U\mathrm{down}_{ij}(t) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

3) Use the motion equation (Eq. 3) to compute $\Delta U\mathrm{up}_{ij}(t)$ and $\Delta U\mathrm{down}_{ij}(t)$.

$$\Delta U\mathrm{up}_{ij}(t) =$$
$$-A[V\mathrm{up}_{ij}(t) + V\mathrm{down}_{ij}(t) - C_{ij}]$$
$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m) f(i,m,j) V\mathrm{up}_{\ell m}(t)$$
$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j) f(\ell,j,m) V\mathrm{up}_{\ell m}(t) \quad (6)$$

$$\Delta U\mathrm{down}_{ij}(t) =$$
$$-A[V\mathrm{up}_{ij}(t) + V\mathrm{down}_{ij}(t) - C_{ij}]$$
$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(\ell,i,m) f(i,m,j) V\mathrm{down}_{\ell m}(t)$$
$$- B \sum_{\ell} \sum_{\substack{m \\ \ell < m}} f(i,\ell,j) f(\ell,j,m) V\mathrm{down}_{\ell m}(t) \quad (7)$$

4) Compute $U\mathrm{up}_{ij}(t + 1)$ and $U\mathrm{down}_{ij}(t + 1)$ on the basis of the first-order Euler method:

$$U\mathrm{up}_{ij}(t + 1) =$$
$$U\mathrm{up}_{ij}(t) + \Delta U\mathrm{up}_{ij}(t)\Delta t \quad (8)$$

$$U\mathrm{down}_{ij}(t + 1) =$$
$$U\mathrm{down}_{ij}(t) + \Delta U\mathrm{down}_{ij}(t)\Delta t \quad (9)$$

where $i = 1,...,N$ and $j = 1,...,N$.

5) Increment t by 1. If $t = T$, then terminate this procedure, otherwise go to step 2.

Consider the nonplanar graph of Jayakumar *et al.* (*2*) in Fig. 3a. The graph has 10 vertices and 22 edges. The latest experiment by Jayakumar *et al.* showed that 19 edges were chosen by their $0(N^2)$ algorithm to construct the maximal planar subgraph as shown in Fig. 3b. Remember that their algorithm does not embed the generated subgraph on a plane but chooses edges for construction of the maximal planar subgraph.

We have developed a simulator based on the proposed algorithm on a Macintosh and on an Apollo 3500 computer. When the coefficients $A = 2$ and $B = 1$ and the unit time $\Delta t = 10^{-5}$ were used for Eqs. 6 through 9, and the initial values of $U\mathrm{up}_{ij}(t)$ and $U\mathrm{down}_{ij}(t)$ were randomized in the range of $-1/10,000$ to 0, the state of our system converged to the global minimum in the 14th iteration. Figure 4, a and b, describes the state of the system at the first and 14th iterations. Our simulator found that the new maximal planar subgraph contains 20 edges instead of 19 edges, which contradicts the result of Jayakumar *et al.* (*2*). Table 1 shows the simulation results where several sets of the coefficients were used. It indicates that either 20 edges or 19 out of 22 edges can be consistently embedded in a single plane.

Within $0(1)$ time the algorithm not only generates a near-maximal planar subgraph from the nonplanar or planar graph but also embeds the subgraph on a plane. On the basis of our observations of the behavior of the simulator, the state of the system always converges to a good solution within 20 or 30 iteration steps. Another simulation result of a graph with 48 vertices and 105 edges also showed the consistency of our algorithm. The algorithm can be implemented by an $N \times N$ two-dimensional neural network array. Among N^2 neurons, only $2M$ neurons are used to obtain the solution. The detailed design of the parallel hardware is given in (*7*).

REFERENCES AND NOTES

1. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).
2. R. Jayakumar *et al.*, *IEEE Trans. CAD* **8**, 3 (March 1989).
3. R. E. Tarjan, *Publication STAN-CS-244-71* (Computer Science Department, Stanford University, Stanford, CA, 1971).
4. H. Szu, in *Proceedings of the International Conference on Neural Networks* (IEEE, Piscataway, NJ, 1989), vol. 2, p. 259.
5. J. Hopfield and D. Tank, *Biol. Cybern.* **52**, 141 (1985).
6. Y. Takefuji in *Proceedings of the International Joint Conference on Neural Networks et al.*, (IEEE, Piscataway, NJ, 1989), p. 529.
7. _____ and K.-C. Lee, *CAISR Tech. Rep. TR 88-139* (Center for Automation and Intelligent Systems Research, Case Western Reserve University, Cleveland, 1988).

30 May 1989; accepted 25 July 1989

Shear Forces in Molecularly Thin Films

Martin Schoen, Charles L. Rhykerd, Jr., Dennis J. Diestler, John H. Cushman

Monte Carlo and molecular dynamics methods have been used to study the shearing behavior of an atomic fluid between two plane-parallel solid surfaces having the face-centered cubic (100) structure. A distorted, face-centered cubic solid can form epitaxially between surfaces that are separated by distances of one to five atomic diameters. Under these conditions a critical stress must be overcome to initiate sliding of the surfaces over one another at fixed separation, temperature, and chemical potential. As sliding begins, a layer of solid exits the space between the surfaces and the remaining layers become fluid.

I N THE LAST DECADE IT HAS BECOME feasible, if not routine, to measure directly on a molecular scale the forces between two solid surfaces separated by a film of fluid (*1–3*). When the distance h between the surfaces is in the range from one to ten "diameters" of the fluid molecule, the component of the force normal to the surfaces oscillates as a function of h, alternating between attraction and repulsion with a period approximately equal to the molecular diameter. This oscillatory character suggests that the fluid near the surfaces arranges itself in layers parallel with surfaces and that entire layers of fluid are successively forced from the space between the surfaces as h is decreased. Statistical mechanics calculations confirm this notion.

Both Monte Carlo (*4–9*) and molecular dynamics (*10–12*) studies of model slit pores (that is, fluid confined between two plane-parallel solid walls) show that an atomic pore fluid does indeed pile up in layers parallel with the walls, whether the walls are structured or not. This "normal" ordering of the fluid is analogous to the ordering of fluid

M. Schoen, Universität Witten/Herdecke, Naturwissenschaftliche Fakultät, Institut für Experimentalphysik, Stockumer Strasse 10, 5810 Witten, Federal Republic of Germany.
C. L. Rhykerd, Jr., and J. H. Cushman, Department of Agronomy, Purdue University, West Lafayette, IN 47907.
D. J. Diestler, Department of Chemistry, Purdue University, West Lafayette, IN 47907.