the classical S–G method in *a priori* elimination of crosswind diffusion. Also, it is superior to the SUPG method owing to an added optimal artificial diffusivity in the streamline direction that substantially improves the stability. The numerical examples support our contentions. Moreover, the generalized S–G method is applicable to other problems.

REFERENCES

[1] D. L. Scharfetter and H. K. Gummel, "Large signal analysis of a silicon read diode oscillator," *IEEE Trans. Electron Devices*, vol. ED-16, pp. 64–77, Jan. 1969.
[2] S. Selberherr, A. Schutz, and H. W. Potzl, "MINIMOS—A two-dimensional MOS transistor analyzer," *IEEE Trans. Electron Devices*, vol. ED-27, pp. 1540–1549, Aug. 1980.
[3] M. Sharma and G. F. Carey, "Semiconductor device simulation using adaptive refinement and flux upwinding," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 590–598, June 1989.
[4] A. N. Brooks and T. J. R. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Comput. Meth. Appl. Mech. Eng.*, vol. 32, pp. 199–259, Sept. 1982.
[5] Y. He, "Discretization method for current continuity equation containing magnetic field," to be published.

Comments on "$O(n^2)$ Algorithms for Graph Planarization"

Yoshiyasu Takefuji, Kuo Chun Lee, and Yong Beom Cho

***Abstract*—This article points out that our parallel algorithm provides the maximum planar subgraph and it is compared with the maximal planar subgraph provided by Jayakumar *et al.* in the above paper. The space–time product complexity is also compared.**

A graph is planar if it can be drawn on a plane with no two edges crossing each other except for their end vertices. The above paper[1] introduced two $O(n^2)$ graph planarization algorithms: PLANARIZE and MAXIMAL-PLANARIZE. The ultimate goal of the algorithms or the graph planarization problem is to maximize the number of edges to be embedded on a plane with no two edges crossing each other except for their end vertices. Although the problem size is small, their algorithm did not provide the global minimum solution for the ten-vertex, 22-edge problem, as shown in Fig. 1. Fig. 2 shows their solution with 19 edges. Our parallel algorithm found the global minimum solution where the maximum planar subgraph has 20 edges for the same problem. Fig. 3 shows our solution.

We are aware that the st-numbering algorithm is used in Jayakumar's algorithm for edge ordering in order to obtain better solutions. However our experiment [1] showed that edge ordering and edge planarization cannot be separated to obtain better solutions. In our algorithm, edge ordering was randomly generated. Although our algorithm is based on randomly generated edge ordering, it always generates the solution with 19 edges or 20 edges.

Manuscript received August 26, 1991. This work was supported by the National Science Foundation through a grant (MIP-8902819) to Y. Takefuji and K. C. Lee.

The authors are with the Department of Electrical Engineering, Case Western Reserve University, Cleveland, OH 44106.

IEEE Log Number 9103794.

[1] R. Jayakumar, K. Thulasiraman, and M. N. S. Swamy, *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 257–267, Mar. 1989.

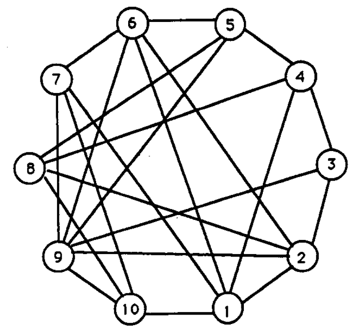

Fig. 1. Nonplanar graph provided by Jayakumar *et al.*

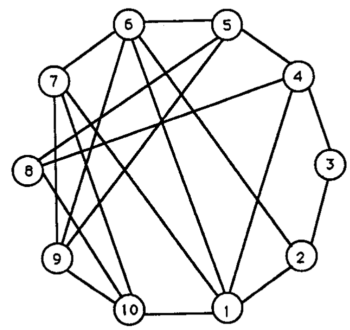

Fig. 2. Maximal planar subgraph provided by Jayakumar *et al.* Three edges, (2, 9), (3, 9), and (2, 8), are removed from the graph in Fig. 1.

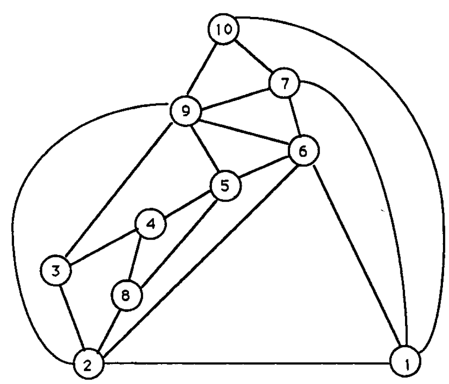

Fig. 3. Maximal planar subgraph using our parallel algorithm. Two edges, (1, 4) and (8, 10), are removed from the graph in Fig. 1.

In our parallel algorithm [1], use is made of $2m$ processing elements that access the $2m$ data elements in parallel, where m is the number of edges. Therefore our algorithm requires $O(m^2)$ space complexity. The number of edges, m, determines the system size (the number of processing elements). In the experiments reported in Table I, in the paper in question, the number of edges, m, is proportional to the number of vertices, $m = O(n)$. This is the usual assumption and it is widely accepted. When we consider space–time product complexity with $m = O(n)$, the complexity of our algorithm is still $O(m^2)$, because the empirical computation time is $O(1)$, where the computation time is determined by the number of iteration steps. $O(1)$ time has been empirically examined by using a variety of nonplanar and planar graphs. In our simulation the solutions can always be obtained within 100 iteration steps. When assuming $m = O(n)$, the space-time product complexity of Jayakumar's algorithm is given by $O(m^3)$, where their space complexity and time complexity are given by $O(m + n)$ and $O(n^2)$ respectively.

When we asume that the target graph is a complete graph with n vertices and $m = n(n - 1)/2$ edges, the complexity of our algorithm is $O(m^2) = O(n^4)$. It is extremely rare in VLSI circuits and printed circuit board routing to assume the condition $m = O(n^2)$ or to deal with the complete graph planarization. The space-time product complexity of Jayakumar's algorithm with $n = O(m^2)$ will be given by $O(n^4)$.

It can be concluded that our algorithm finds the maximal solution in $O(1)$ time with $O(m)$ processing elements and the space-time product complexity is given by $O(m^2)$, while Jayakumar's algorithm requires $O(n^2)$ time with a single processor and their space-time product complexity is given by $O(m^3)$. One should know that our algorithm not only generates a maximal planar subgraph but also embeds the subgraph on a single plane in $O(1)$ time with $O(m^2)$ space-time product complexity, while Jayakumar's algorithm only generates a maximal planar subgraph in $O(m^2)$ time with $O(m^3)$ space-time product complexity when assuming $m = O(n)$. The space-time product complexity of our algorithm is given by $O(n^4)$ while that of Jayakumar's algorithm is given by $O(n^4)$ when assuming $m = O(n^2)$.

References

[1] Y. Takefuji and K. C. Lee, "A near-optimum parallel planarization algorithm," *Science*, vol. 245, pp. 1221-1223, Sept. 1989.