Discrete Applied Mathematics 25 (1989) 317-320
North-Holland

NOTE

ON THE ORDERABILITY PROBLEM FOR PLA FOLDING*

S.S. RAVI
Department of Computer Science, SUNY at Albany, 1400 Washington Avenue, Albany, NY 12222, USA

Received 29 June 1987
Revised 26 July 1988

An interesting graph theoretic problem, called the orderability problem for programmable logic array (PLA) folding, was formulated and shown to be NP-complete. We show that a closely related problem, called the orderability problem for bipartite folding, is solvable in linear time.

1. Definitions and problem formulation

We consider an interesting graph theoretic problem that arises in conjunction with PLA folding [2,5,7]. This problem involves *mixed* graphs, that is, graphs that contain both directed and undirected edges. Given a directed edge (x,y), we refer to the nodes x and y as *tail* and *head* nodes respectively. Using this terminology, the problem can be formulated as follows.

Orderability for bipartite folding (OBF)

Instance: An undirected graph $G(V,E)$ and a set $F=\{\langle p_i,q_i\rangle\colon 1\le i\le r\}$ of undirected edges none of which is in E and which form a matching.[1]

Question: Can the edges of F be oriented in such a way that for every tail node t and every head node h, the edge $\{t,h\}$ is *not* in E?

For a given OBF instance, any orientation of F that satisfies the above condition is called a *valid* orientation; if there is a valid orientation, the set F is said to be *orientable*.

In order to place the above problem in proper perspective, we use some definitions from [5]. An *alternating path* in a mixed graph is a simple path whose first and last edges are directed edges[2] and in which directed and undirected edges alternate. An *alternating cycle* in a mixed graph is an alternating path such that there

* Research supported in part by the National Science Foundation under grants MCS-8103713 and DCI-8603318.

[1] A set of edges forms a *matching* if no two edges in the set are incident on the same node.

[2] Each directed edge must be traversed along its direction.

is an undirected edge between the initial and the terminal nodes of the path. The reader can easily verify that the OBF problem defined above is equivalent to the problem of determining whether the edges of F can be oriented in such a way that the resulting mixed graph does *not* have an *alternating path*. We will show in Section 2 that this problem can be solved in linear time. In contrast, the *orderability problem for PLA folding* (OF) [5], in which we are required to determine whether the edges in F can be oriented in such a way that the resulting mixed graph does *not* have an *alternating cycle*, is NP-complete[3] [5]. Thus although the specifications of OBF and OF problems are very similar, their complexities are very different if $P \neq NP$.

2. A linear time algorithm for the OBF problem

Throughout this section, we use standard graph theoretic terminology [6]. We begin our discussion with the observation that the nodes of G which do not appear in any edge of F have no role in determining the orientability of F and so may be deleted from G. We therefore assume that each node of G appears in some edge of F. With this assumption, the following (easily verified) lemma points out a relationship between connected components of G and the orientability of F. This lemma plays a key role in our algorithm.

Lemma 1. *Let an undirected graph $G(V,E)$ and set F constitute an instance of the OBF problem with each node of G appearing in some edge of F. Then, in any valid orientation of F, each connected component of G must be composed entirely of head nodes or of tail nodes.*

The above lemma indicates that for purposes of determining orientability, we can treat each connected component of G as a single unit. To understand the role played by this lemma, let us consider an edge $\langle p_i, q_i \rangle$ in F. In any valid orientation of F, one of p_i and q_i must be a head node and the other a tail node. Thus Lemma 1 implies that the nodes p_i and q_i must be in *different* connected components of G (i.e., if for some edge $\langle p_j, q_j \rangle$ in F, p_j and q_j are both in the same connected component of G, then F is not orientable). Suppose there is a valid orientation of F with p_i as a tail node and q_i as a head node. If p_i is in the connected component C_1 and q_i is in C_2, then by Lemma 1, *all* of the nodes in C_1 must be tail nodes and *all* of the nodes in C_2 must be head nodes. Therefore, every edge in F induces constraints on a pair of connected components of G.

To handle these constraints, we construct an auxiliary undirected graph, $G_c(V_c, E_c)$.[4] Each node in V_c corresponds to a connected component of G. There is

[3] For NP-completeness and related topics, see [4].

[4] We will permit "self loops" in G_c.

an edge between nodes x and y of V_c if and only if for some edge $\langle p_i, q_i \rangle$ in F, p_i is in the connected component corresponding to x and q_i is in the connected component corresponding to y. The following theorem shows how the graph G_c can be used to determine the orientability of F.

Theorem 2. *Let the undirected graph $G(V,E)$ and set F constitute an instance of the OBF problem such that each node of G appears in some edge of F. Then, F is orientable if and only if the auxiliary graph $G_c(V_c,E_c)$ obtained from G and F is bipartite.*

Proof. ($\Leftarrow$) Suppose G_c is bipartite. Let V_{c1} and V_{c2} represent a bipartition of V_c. We obtain an orientation of F by assigning all of the nodes of G which are in the connected components corresponding to the nodes in V_{c1} as head nodes and all of the nodes which are in the connected components corresponding to the nodes in V_{c2} as tail nodes. It is easy to verify that this is a valid orientation of F.

($\Rightarrow$) Suppose F is orientable. Consider any valid orientation of F. In that orientation, each connected component of G must be composed entirely of tail nodes or of head nodes (Lemma 1). Therefore, we can assign (unambiguously) a label T or H to each node x of the auxiliary graph G_c depending upon whether the connected component of G corresponding to x is composed of tail nodes or of head nodes. It is easy to verify that this labeling provides a 2-coloring of G_c. Thus G_c is bipartite. □

Theorem 2 leads directly to an algorithm for the OBF problem.

Algorithm Orient

Input. Undirected graph $G(V,E)$ and set $F=\{\langle p_i,q_i\rangle: 1 \le i \le r\}$.

Output. A valid orientation of the edges of F, if one exists.

```
begin
  Obtain the connected components of G;
  Construct the auxiliary graph Gc(Vc,Ec);
  if Gc is bipartite
  then Orient the edges in F according to the bipartition
  else print "there is no solution";
end
```

We conclude by pointing out that the above algorithm can be implemented to run in $O(|V|+|E|)$ time. It is well known that the connected components of a graph $G(V,E)$ can be obtained in $O(|V|+|E|)$ time using depth-first search [1]. The auxiliary graph G_c contains at most $|V|$ nodes and at most $|F|=O(|V|)$ edges (because each pair in F adds one edge to E_c). Testing whether G_c is bipartite can be done using breadth-first search in $O(|V_c|+|E_c|)=O(|V|)$ time [3]. Thus the overall running time of the algorithm is $O(|V|+|E|)$.

Acknowledgment

I would like to express my sincere thanks to the referees for their valuable suggestions. I am indebted to my thesis advisor Prof. Errol L. Lloyd (University of Pittsburgh) for several helpful discussions and for his comments on the earlier drafts of this paper.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, The Design and Analysis of Algorithms (Addison-Wesley, Reading, MA, 1974).
[2] J.R. Egan and C.L. Liu, Optimal bipartite folding and partitioning of a PLA, IEEE Trans. CAD Integrated Circuits and Syst. 3(3) (1984) 191-199.
[3] S. Even, Graph Algorithms (Computer Science Press, Rockville, MD, 1979).
[4] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, San Francisco, CA, 1979).
[5] G.D. Hachtel, A.R. Newton and A.L. Sangiovanni-Vincentelli, An algorithm for optimal PLA folding, IEEE Trans. CAD Integrated Circuits and Syst. 1(2) (1982) 63-77.
[6] F. Harary, Graph Theory (Addison-Wesley, Reading, MA, 1969).
[7] S.S. Ravi and E.L. Lloyd, On the complexity of near-optimal PLA folding, Tech. Rept. 84-18, Department of Computer Science, SUNY at Albany, NY (1984); also: SIAM J. Comput. 17 (1988) 696-710.