Approximation Algorithms for PLA Folding

by

Teofilo F. Gonzalez
Programs in Computer Science
The University of Texas at Dallas
Richardson, Texas 75080

Abstract:

In this paper we consider the problem of PLA folding. As input we are given a PLA that implements a set of functions. It is required to transform this PLA into a minimal equivalent PLA, i.e., an equivalent PLA with the minimal number of columns. It is well known that this problem is NP-hard and several authors have studied algorithms that obtain suboptimal solutions to this problems. All the known approximation algorithms have worst case approximation bound that is not better than $\lfloor n/3 \rfloor$, where n is the number of columns in the PLA. In this paper we present a series of approximation algorithms that guarantee solutions that are within a bound of $\lfloor n/2k \rfloor$, where k is any fixed constant. Our algorithms have time complexity $O(n^{2k})$.

Keywords: PLA folding, Approximation algorithms for NP-hard problems, VLSI design.

I. INTRODUCTION

In [HNS] the problem of optimally folding a PLA was reduced to a graph problem in which it is required to insert the maximum number of directed edges into an undirected graph without introducing a special type cycle. More formally, we are given an undirected graph, G=(V,E), called the intersection graph (see [HNS]). From this graph we construct a mixed graph, G', by adding to G a set of directed edges. The problem consists of constructing a mixed graph G' with the maximum number of directed edges without violating the following conditions:

a) if (n_i, n_j) is a directed edge in G' then $\{n_i, n_j\} \notin E$;

b) no node in V is incident to or from more than one directed edge; and

c) there is no alternating cycle in the mixed graph, i.e., graph G' does not have a cycle of even length such that when traversing the edges in the cycle the directed and undirected edges alternate and all the directed edges are oriented in the same direction.

In [HNS] it was shown that this problem is NP-hard and several heuristics for its solution were introduced. Recently, Ravi and Lloyd [RL] showed that all the heuristics in [HNS] behave in the worst case rather poorly. It was shown that $(f^*/f) \leq \lfloor n/3 \rfloor$ for the best of these heuristics, where n is the number of columns, f is the number of columns folded by the approximation algorithm and f^* is the number of columns folded in an optimal solution. Furthermore, it was shown that this bound is best possible, i.e., there are examples that achieve this ratio. All the approximation algorithms are of polynomial time complexity, with the polynomial having both a small constant and exponent.

In this paper it is shown that for any positive constant, k, greater than one there is an $O(n^{2k})$ algorithm that guarantees solutions with worst case ratio $\leq \lfloor n/2k \rfloor$. This bound can be shown to be best possible. It should be pointed out that our algorithm is useful only when k is small.

II. ALGORITHMS

In this section we present an algorithm with worst case time complexity $O(n^{2k})$, where k is any positive integer, that guarantees solutions with worst case ratio $\leq \lfloor n/2k \rfloor$. The strategy used by our algorithm is "brute force". Starting from G we add all subsets of at most k edges to it without violating condition (a) in the definition of a mixed graph. Our solution is one of the mixed graphs generated with the maximum number of directed edges in it.

```
ALGORITHM BRUTE( k , G );
   for ℓ = k to 1 by -1 do
    begin
     // every edge e ∉ E is said to be missing in G //
     for every subset S of missing edges in G
         such that |S| = ℓ do
      begin
       for every partition of S into S1 and S2 do
        begin
         Let G' = G and add to G' all the directed edges
           (ni , nj) such that ( {ni , nj} ∈ S1
           and i < j ) or ({ni , nj} ∈ S2 and i > j);
         if G' is a mixed graph then output (G') and stop
         endif
       end
     end
   end
end of algorithm
```

<u>Theorem 1</u>: Algorithm BRUTE guarantees solution with worst case approximation ratio $\leq \lfloor n/2k \rfloor$.

<u>Proof</u>: It is simple to see that our algorithm generates an optimal solution whenever there is one whose value is $\leq k$. If this is not the case our algorithm will obtain an optimal solution that introduces exactly k directed edges. The value of the optimal solution is at most $\lfloor n/2 \rfloor$. Hence, the approximation bound is

$\leq \lfloor n/2k \rfloor$.   []

Theorem 2: The time complexity of algorithm BRUTE is $O(n^{2k})$.

Proof: The proof is simple and will be omitted. Note that k is a fixed constant independent of n.   []

III REFERENCES

[HNS] Hatchel, G.D., A. R. Newton and A.L. Sangiovanni-Vincentelli, "Techniques for Programmable Logic Array Folding," Proceedings of the 19th Design Automation Conference, June 1982, 147-155.

[RL] Ravi, S. S. and E. L. LLoyd, "Analysis of Heuristics for PLA Folding," Proceedings of the 20th Allerton Conference on Computing, Communications and Control, Nov 1982, 646-655.