DIMACS - Center for Discrete Mathematics & Theoretical Computer Science

RUTCOR - Rutgers Center for Operations Research

Workshop on Applications of

Combinatorial Optimization in Science and Technology

COST

April 2-6, 1991

A graph theoretic approach for PLA area optimization

A. G. Ferreira*
Laboratoire de l'Informatique du Parallélisme - CNRS
École Normale Supérieure de Lyon - France
e-mail: ferreira@lip.ens-lyon.fr

S. W. Song†
IME - Department of Computer Science
University of São Paulo - Brazil
e-mail: song@brusp.ansp.br

Extended Abstract

1 Introduction

The Programmable Logic Array (PLA) is a regular layout of logical circuits that is commonly used to implement multiple-output combinatorial logic circuits. Logical functions in the canonical form of sum of products can easily be implemented by a PLA. As we shall see later, a PLA consists of a number of rows and columns. At each intersection of a row with a column a logical circuit can be placed. The number of logical circuits actually placed is often very small relative to the total number of row-column intersections. PLA *folding* is a technique that, given a PLA, attempts to produce an equivalent folded PLA that occupies less space. The PLA folding problem has been extensively discussed in the literature [7, 8, 9].

A PLA is represented by a 0-1 personality matrix where a 1 denotes the presence of a logical circuit. We assume $n = \max(r, c)$ for a PLA personality matrix with r rows and c columns. The PLA folding problem attempts to obtain a compaction of the columns after a suitable row permutation. The optimal multiple/simple PLA folding problem has been proven to be NP-complete [10]. We present $O(n^3)$ heuristic algorithms for multiple/simple folding and give a sufficient condition for the solution to be optimal. Verification of the sufficient condition constitutes part of the proposed algorithms, and is in fact used to obtain the solution. The sufficient condition for optimality is that a certain submatrix of the personality matrix, called its reduced personality

*On leave from the Institute of Mathematics and Statistics of the University of São Paulo and member of Project 30.01 of the Program BID/USP.

†Supported by the Program BID/USP - Project 30.01, FAPESP - Proc. No. 88/3095-9 and CNPq - Proc. No. 306063/88-3. This paper was written during a visit of the second author to LIP/ENS Lyon.

matrix, possesses the consecutive ones property for columns. The relation between this layout problem and matrices with the consecutive ones property has been established independently by Deo, Krishnamoorthy and Langston [2] and Ferreira [4]. Certain matrix classes have been shown in [13] that do not possess the consecutive ones property. We show that the class of personality matrices that do not satisfy the sufficient condition for optimality is strictly smaller than the class of matrices that do not possess the consecutive ones property. We also discuss some lower bounds of the optimal solution, based on the maximum number of 1's in each row, the size of the maximum clique and the chromatic number of the associated column adjacency graph.

Many heuristic algorithms for the multiple/simple PLA folding problems have been proposed. Performance studies of such heuristic algorithms are based only on test runs on experimental data. Lecky et al. [9] considered disjoint column-pair compatibility graphs where cliques correspond to PLA folding sets. Near-optimal cliques can often been obtained by their proposed heuristic algorithms of $O(n^4)$ and $O(n^5)$. Performance evaluation was done through numerous test runs and experimental results show the proximity of the obtained results to the optimal solution, which was known beforehand. Hwang et al. [8] proposed $O(n^3)$ and $O(n^4)$ heuristic algorithms based on longest paths on the associated disjoint graph generated from the personality matrix. Through experimental testing, they compared their work with that of Hachtel et al. [7]. See [9] for a brief history of the development of PLA folding heuristics.

We present in the following a summary of our results, and refer the interested reader to [5].

2 The PLA folding problem

In a PLA (Programmable Logic Array) the input signals run, say vertically, through a matrix of circuit elements called the *AND plane.* Combinations of the inputs and their complements are selected in this AND plane. These signals then run horizontally and become inputs to another matrix of circuit elements called the *OR plane.* The outputs of the OR plane are the canonical sum-of-products form of Boolean functions of the PLA inputs.

Personality matrix Consider input variables v_1, v_2, ..., v_k and output functions f_{k+1}, f_{k+2}, ..., f_c with a total of r product terms. The corresponding PLA can be represented by a $r \times c$ *personality matrix* $M = (m_{ij})$, a (0,1)-matrix defined as follows.

$$\text{for } 1 \le i \le r, 1 \le j \le c, m_{ij} = \begin{cases} 1 & \text{if a logical circuit is present at } v_j, \bar{v_j} \text{ or } f_j \\ 0 & \text{otherwise} \end{cases}$$

The columns of matrix M will be called *logical columns.*

For a logical column j of a given configuration $M' = \left(m'_{ij}\right)$, obtained by row permutation of M, we denote by $\text{Int}(M', j)$ the interval $[a, b]$, where

$$\begin{aligned} a &= \min i \text{ such that } m'_{ij} = 1 \text{ and} \\ b &= \max i \text{ such that } m'_{ij} = 1. \end{aligned}$$

We say that a subset of logical columns of M', denoted by s_j,

$$s_j = \{j_1, j_2, \ldots, j_m\}$$

is *foldable* to a single *physical column* j if and only if

$$\text{Int}(M', j_1) \cap \text{Int}(M', j_2) \cap \ldots \cap \text{Int}(M', j_m) = \phi.$$

The reason behind this restriction is to avoid electrical signal interference when implementing the logical circuits of the PLA.

Given a personality matrix M of zeros and ones, we want to find the configuration, among all the row permutations of the given matrix, whose columns can be foldable to the minimum number of physical columns or subsets s_j. The simple PLA folding problem adds the constraint of not folding more than two logic columns in one physical column.

For each row, we can define an associated set whose elements are the column positions where we have 1's. As in [2], let us first remove all duplicate rows, so that only distinct rows remain. For the remaining rows, a row is said to be *dominant* if its associated set of column positions with 1's is not a proper subset of those associated with any other row. A row that is not dominant is said to be *dominated.* We will now proceed to remove all dominated rows so as to obtain a final matrix, which we call the *reduced personality matrix* or simply *reduced matrix*, with only dominant rows. As noted in [2], any given personality matrix can be folded using the same number of physical columns as its reduced matrix. The dominated (resp. duplicate) rows can be repositioned next to their corresponding dominant (resp. original) rows.

Hence, for the discussions from now on, we consider the personality matrix already in the reduced form, unless explicitly stated. The cost for removing rows and their subsequent repositioning is $O(n^3)$.

Consider a $r \times c$ personality matrix M and let $n = \max(r, c)$. We divide the optimal PLA folding problem into two steps, namely,

1. Permutation step: find the permutation of the rows which leads to the minimum number of subsets s_j. The exponential nature of the problem is due to this step.

2. Compaction step: compact the matrix obtained by step 1 by merging its logical columns into physical columns. This step is solvable in polynomial time.

We will use the following terminology, according to Paillottin [12].

Gate: the part of a logical column between (and including) the two extreme 1's.

Peak function: For a given configuration of matrix M, the peak of a row is equal to the number of gates it crosses.

$P*$: the maximum of the peak function taken over all the rows.

3 Multiple PLA folding

We first show that the optimal compaction for a given matrix configuration can be done in $O(n \log n)$. Then we discuss the permutation step.

3.1 Compaction step: $O(n \log n)$

For a given row permutation of matrix M, given by the permutation step, the compaction step assigns the logical columns to new physical columns.

The use of a greedy $O(n^2)$ algorithm for this step was proved to be optimal [12], positioning the gates in exactly $P*$ physical columns. In [5], we show how such an algorithm can be implemented in $O(n \log n)$ by organizing the input in a balanced search tree.

3.2 Complexity results

In Section 3.1 we determine the exact number of necessary physical columns for a given matrix configuration. There is at least one row permutation of matrix M that yields the best possible compaction, i.e., the one which leads to the minimum $P*$, denoted by $P*_{\min}$. In this way any compacted configuration will need $P*_{\min}$ or more physical columns to be implemented.

Let Ω be the function that gives for each row, the number of 1's it crosses. Let $\Omega*$ be the maximum of Ω computed over all the rows. Note that Ω and $\Omega*$ are characteristic of a given matrix, independent of any permutation of its rows.

Proposition 3.1 *For any given personality matrix M, $\Omega* \leq P*_{\min}$, i.e., $\Omega*$ is a lower bound for any multiple folding carried out in the given PLA.*

Corollary 3.1 *A folding that positions the logical columns of M in $\Omega*$ physical columns is optimal.*

Corollary 3.2 *A row permutation of the personality matrix M in which the gates have solely 1's is an optimal configuration, meaning that it leads to an optimal folding, since $P* = \Omega*$.*

3.3 Permutation step: linear in the size of input

We have the following theorem, which follows directly from a result of [1].

Theorem 3.1 *Given a $r \times c$ personality matrix M with d non-zero entries, if it has the consecutive ones property for columns, then we can get its optimal configuration in $O(r+c+d)$, i.e., at most $O(n^2)$ time.*

The sufficient condition for optimality is that matrix M meets the condition of Theorem 3.1. We show in [5] that the class of personality matrices that do not satisfy the sufficient condition for optimality is strictly smaller than the class of matrices that do not possess the consecutive ones property.

3.4 Heuristic algorithm for multiple folding

If the personality matrix possesses the consecutive ones property, then Booth and Lueker's algorithm ([1]) gives the best row permutation to achieve optimal compaction. Otherwise, the algorithm would simply stop, reporting the failure of finding the consecutive ones property. In both cases, when the algorithm stops, a row permutation will be available in which every processed column so far has its 1's arranged consecutively. What we propose below is to utilize their algorithm to produce an acceptable permutation, even if the sufficient conditions for optimality are not verified.

The algorithm to be proposed in this section attempts to get a good permutation by minimizing the distance between the extreme 1's, generating small gates.

Each column yields a subset whose elements are the row positions with 1 in it. These subsets are called restriction subset in the sense that we want all the row positions of the subset to be consecutive in the final configuration.

Our heuristic attempts to increase the number of processed columns. Hence, one immediate modification is to change the termination strategy in Booth and Lueker's algorithm, so that all the restriction subsets are examined instead of stopping as soon as an unsatisfied subset is found. Then several heuristics are possible.

Some examples showed that the strategy consisting of sorting the restriction subsets in descending order is superior to the ascending order strategy especially for dense matrices. We note that since this step is relatively "cheap", we might as well use both heuristics separately and choose the one that gives the better solution, i.e., the one that leads to the least $P*$.

We will call what we have discussed above the modified Booth and Lueker's algorithm.

Heuristic algorithm for multiple folding.

a.	read in the personality matrix, not in reduced form	$O(n^2)$
b.	remove duplicate and dominated rows	$O(n^3)$
c.	run modified Booth and Lueker's algorithm on the reduced matrix	$O(n^2)$
d.	rearrange the matrix	$O(n^2)$
e.	reinsert the rows removed at step b	$O(n^2)$
f.	compact the matrix by Algorithm 2.2	$O(n \log n)$
g.	output the folded matrix	$O(nP*)$

4 Simple PLA folding

In the simple PLA folding problem, we position at most two logical columns on each physical column. A matrix with n columns has an obvious lower bound of $n/2$ on the number of physical columns of the folded PLA. Most of the ideas of the multiple folding problem still apply here, and the complexity of the heuristic algorithm for simple PLA folding is $O(n^3)$ as well.

We notice, however, some differences. The greedy algorithm used for the compaction step of the multiple folding does not apply for the simple folding. We use then a maximum matching algorithm [11], since only two logical columns can share the same physical column in the simple folded PLA.

As a consequence, the proof that the consecutive ones permutation is optimal was no longer valid. Nevertheless, such result is still true but it has to be proven in a different way [5]. As for the proposed heuristic, it is basically the same as the one introduced for the multiple folding. The only difference lies in the compaction step where the greedy algorithm is replaced by a maximum matching algorithm on an associated gate compatibility graph.

Acknowledgments:

The second author would like to thank Professor M. Cosnard for his invitation to visit LIP/ENS Lyon (Laboratoire de l'Informatique du Parallélisme) and all the LIP members for their hospitality.

References

[1] K. S. Booth and G. S. Lueker, "Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms", *Journal of Computer and System Sciences* 13, pp. 335-379, 1976.

[2] N. Deo, M. S. Krishnamoorthy and M. A. Langston, "Exact and approximate solutions for the gate matrix layout problem", *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 79-84, January 1987.

[3] M. R. Fellows and M. A. Langston, "Nonconstructive advances in polynomial-time complexity", *Information Processing Letters*, 26, pp. 157-162, 1987.

[4] A. G. Ferreira, "An Optimal $O(n^2)$ algorithm to fold special PLA's", in *Lecture Notes in Economics and Mathematical Systems*, No. 302, Proceedings of the Conference "Optimization Days 86", Montreal, Canada, April 30 - May 2, 1986, H. A. Eiselt and G. Pederzoli (eds.), pp. 92-102, Springer-Verlag, 1988.

[5] A. G. Ferreira and S. W. Song, "Achieving optimality for gate matrix layout and PLA folding: a Graph Theoretic Approach", Rapport de Recherche, LIP/ENS Lyon, in print.

[6] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, 1980.

[7] G. D. Hachtel, A. R. Newton and A. L. Sangiovanni-Vicentelli, "An Algorithm for optimal PLA folding", *IEEE Trans. Computer-Aided Design*, vol. CAD-1, pp. 63-76, January 1982.

[8] S. Y. Hwang, R. W. Dutton and T. Blank, "A best-first search for optimal PLA folding", *IEEE Trans. Computer-Aided Design*, vol. CAD-5, pp. 433-442, July 1986.

[9] J. E. Lecky, O. J. Murphy and R. G. Absher, "Graph-Theoretic Algorithms for the PLA folding problem", *IEEE Trans. Computer-Aided Design*, vol. CAD-8, pp. 1014-1021, September 1989.

[10] M. Luby, U. Vazirani, V. Vazirani and A. Sangiovanni-Vicentelli, "Some theoretical results on the optimal PLA folding problem", in *Proceedings IEEE Conf. on Circuits and Computers*, pp. 165-170, 1982.

[11] S. Micali and V. V. Vazirani, "An $O(|V|^{1/2}|E|)$ algorithm for finding maximum matching in general graphs", *Proceedings 21st Annual Symposium on Foundations of Computer Science*, IEEE, pp. 17-27, 1980.

[12] J. F. Paillotin, "Optimization of the PLA area", *Proceedings 18th Design Automation Conference*, pp. 406-410, June 1981.

[13] A. C. Tucker, "A structure theorem for the consecutive ones property", *Journal Combinatorial Theory* 12(B), pp. 153-162, 1972.