

**First Steps Towards Experimental Design in Evaluating Layout
Layout Algorithms: Wire Length versus Wire Crossing in
Linear Placement Optimization**

Debabrata Ghosh Franc Brglez Matthias Stallmann

October 1998

1998-TR@CBL-11-Ghosh, Version 1.0

Reprinted, with permission, from
<http://www.cbl.ncsu.edu/publications/>

Publications at this site are occasionally revised,
please check for the latest version under the same title.

For more information about CBL, visit
<http://www.cbl.ncsu.edu/>

or write to
info@cbl.ncsu.edu

ABOUT THIS DOCUMENT

Acknowledgments. F. Brglez and D. Ghosh have been supported by contracts from the Semiconductor Research Corporation (94-DJ-553), SEMATECH (94-DJ-800), and DARPA/ARO (P-3316-EL/DAAH04-94-G-2080 and DAAG55-97-1-0345), and a grant from Semiconductor Research Corporation.

Abstract. This report introduces an experimental design that discovers a new relationship between a cell placement for minimized wire crossing in bipartite (two-layer) graphs and a cell placement in linear arrangement, optimized for minimum total wire length as measured *after rectilinear routing* in a single channel. We introduce *hypercrossing*, a new crossing model for graphs, and demonstrate that the total wire length in the channel, as measured after routing, has a positive linear correlation with the total hypercrossing, and a coefficient of correlation of near 100%. While heuristics to directly minimize total hypercrossing are yet to be invented, our experiments show that the minimization of total (regular) wire crossing leads to placements with significantly reduced wire length and wiring area.

A methodology and new datasets are now available to determine, *with statistical significance*, which of the two cost function classes will yield a better placement in general: costs based on the model of the wire length, or costs based on the model of the wire crossing. Our best results to date have been achieved with a wire crossing cost model.

Keywords. linear placement problem, cell placement optimization, wire length and wire crossing minimization graph equivalence classes, design of experiments

Note. This document has been published for viewing on the Web in Postscript and HTML formats. The latter is annotated with active hyperlinks to facilitate quick browsing of this and related documents.

Citation. If you choose to cite this report, please add the following entry to your bibliography database:

```
@techreport{
  1998-TR@CBL-11-Ghosh,
  author = "D. Ghosh and F. Brglez and M. Stallmann",
  title = "{ First Steps Towards Experimental Design in Evaluating
  Layout Algorithms: Wire Length versus Wire Crossing in Linear Placement
  Optimization }",
  institution = "{CBL, CS Dept., NCSU, Box 7550, Raleigh, NC 27695}",
  number = "1998-TR@CBL-11-Ghosh",
  month = "October",
  year = "1998",
  note = "{Also available at http://www.cbl.ncsu.edu/publications}"
}
```

CONTENTS

| | | |
|------------|-----------------------------------|----------|
| I | Introduction | 2 |
| II | Background and Motivation | 2 |
| III | Design of Experiments | 4 |
| IV | Equivalence Classes | 5 |
| V | Experimental Results | 6 |
| VI | Conclusion and Future Work | 8 |
| VII | Appendix | 9 |

LIST OF FIGURES

| | | |
|----|---|----|
| 1 | Layouts and corresponding graph for netlist instance <code>ref_multi1_16-76-0001</code> : (a) a section of a layout corresponding to <code>tr20</code> , (b) a section of a layout corresponding to <code>tr19</code> , (c) graph of the layout in (a), (d) graph of the layout in (b). | 3 |
| 2 | Design of experiments to compare the placement produced by different algorithms (treatments). | 5 |
| 3 | Workflow for treatments and typical correlations. | 6 |
| 4 | Wire length in layout, as reported by the router, for Treatments 0-20 for two equivalence classes. | 7 |
| 5 | Wire length in layouts versus wire hypercrossing and regular crossing in graphs. | 8 |
| 6 | Asymptotic behavior of treatments 19 and 20. | 9 |
| 7 | Correlations between wire length in layout and regular crossing, wire length and hypercrossing and regular crossing and hypercrossing. | 10 |
| 8 | First level of the MCNC benchmark <code>cc.blif</code> that is used for generating the graph equivalence classes <code>cc_level_n</code> | 11 |
| 9 | Wire length in layout versus hypercrossing and regular crossing and correlation between hypercrossing and regular crossing. | 11 |
| 10 | Wire length in layout versus hypercrossing and regular crossing. | 12 |
| 11 | Asymptotic behavior of treatments 19 and 20 for <code>cc_level_n</code> equivalence classes. | 12 |

First Steps Towards Experimental Design in Evaluating Layout Algorithms:

Wire Length versus Wire Crossing in Linear Placement Optimization

Debabrata Ghosh

Franz Brglez

Matthias Stallmann

CBL (Collaborative Benchmarking Lab), Dept. of Comp. Science, Box 7550, NC State U., Raleigh, NC 27695
<http://www.cbl.ncsu.edu/>

Abstract – *This paper introduces an experimental design that discovers a new relationship between a cell placement for minimized wire crossing in bipartite (two-layer) graphs and a cell placement in linear arrangement, optimized for minimum total wire length as measured after rectilinear routing in a single channel. We introduce hypercrossing, a new crossing model for graphs, and demonstrate that the total wire length in the channel, as measured after routing, has a positive linear correlation with the total hypercrossing, and a coefficient of correlation of near 100%. While heuristics to directly minimize total hypercrossing are yet to be invented, our experiments show that the minimization of total (regular) wire crossing leads to placements with significantly reduced wire length and wiring area.*

A methodology and new datasets are now available to determine, with statistical significance, which of the two cost function classes will yield a better placement in general: costs based on the model of the wire length, or costs based on the model of the wire crossing. Our best results to date have been achieved with a wire crossing cost model.

Keywords: cell placement optimization, wire length and wire crossing minimization.

I. INTRODUCTION

Area and performance in submicron-technology VLSI circuits are critically dominated by the interconnect. The variations in interconnect that arise *after* cell placement and routing, such as wire crossings, wire density, wire loading, wire length, etc., may be hard if not impossible to predict. Similarly, the interconnect models associated with the logic implementation and logic testability *prior* to placement and routing may be subject to considerable variations.

In row-based cell placement, the cost function that continues to dominate the research relates mostly to total wire length. Algorithms that use wire length as a cost function include approaches as diverse as simulated annealing [1, 2], quadrisection [3], and recursive and quadratic programming [4, 5, 6, 7, 8, 9, 10]. Comparisons between different wire length objectives are presented in [11, 12]. Models and algorithms for accurately estimating the average interconnect length for both random and optimized placements have been presented in [13]. Abstract placement models, such as presented in [14], also aim

to minimize the channel density.

In contrast, there have been few algorithms based on the wire crossing cost function. The crossing number and wire area have been investigated to find lower bounds on the layout area and the maximum edge length of a variety of computational networks [15]. Given a global routing, the problem of ‘properly’ distributing the set of crossings among the regions has been studied in [16]. Crossing minimization as a post-optimization problem after global routing is reported in [17].

However, in fields as diverse as economics, social sciences, mathematics and computer science, research on crossing number minimization in graphs has been carried out for many years. Crossing theory has been developed to improve the readability of hierarchical structures [18]. The problem of placing vertices for minimum crossing number is NP-complete, even for 2-layer graphs [19]. Graph drawing packages can now be obtained from the Web [20], [21]. A textbook, bringing this research up-to-date, is now available [22].

This paper introduces *hypercrossing*, a new crossing model for graphs, and a series of experimental designs that demonstrate, *with statistical significance*, important relationships between the cell placement for minimized (regular) wire crossing in bipartite (two-layer) graphs, and cell placement in linear arrangement, optimized for minimum total wire length as measured *after rectilinear routing* in a single channel:

- The total wire length in the channel as measured *after routing* has a positive linear correlation with the total hypercrossing, and the coefficient of correlation of near 100%, *regardless of the choice of placement algorithm*. This suggests that minimizing total hypercrossing also minimizes the total wire length and vice versa.
- While heuristics to minimize total hypercrossing are yet to be invented, 18 out of 19 placement algorithms whose objective is to minimize (regular) wire crossing achieve a better minimum on average (of wire crossing and wire length) [23] than a state-of-the-art placement algorithm whose objective is to minimize the total wire length [24].
- An equivalence class of two-layer graphs that are all isomorphic to each other (but presented in different random orders) can induce nearly as much variance in the optimized wire length (and wire crossing) as does an equivalence class of graphs of the same size but with random interconnection and fanout.

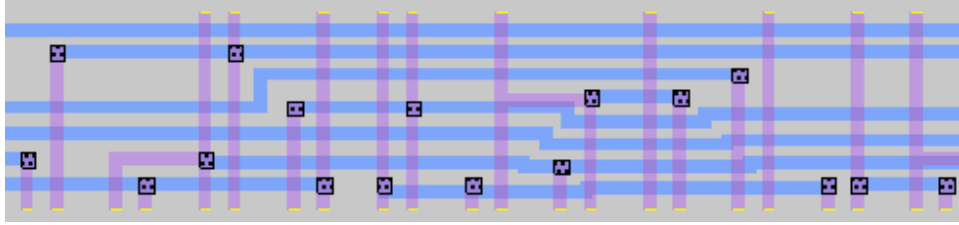
The paper is organized into following sections: (2) Background and Motivation; (3) Design of Experiments; (4) Bigraph Equivalence Classes; (5) Experimental Results; (6) Conclusions.

II. BACKGROUND AND MOTIVATION

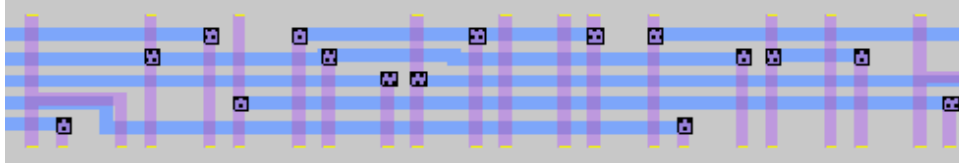
This work is motivated by a series of experiments with standard cell placement and routing we conducted earlier. Briefly, we have observed a high degree of correlation between the total wire length and the total wire crossing as reported by two tools that have very different layout objectives: a place and route tool [24] and a graph drawing tool [20]. The objective of the first tool is to place a netlist of standard cells into rows and minimize the area of the layout *after routing*: total area and total wire length

* F. Brglez and D. Ghosh have been supported by contracts from the Semiconductor Research Corporation (94-DJ-553), SEMATECH (94-DJ-800), DARPA/ARO (P-3316-EL/DAAH04-94-G-2080), and (DAAG55-97-1-0345), and a grant from Semiconductor Research Corporation.

“Permission to make digital/hard copy of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of CBL. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.”



(a) Total wire length reported by router = $16,726\mu$, total routing area reported by router = $338,670 \mu^2$



(b) Total wire length reported by router = $13,648\mu$, total routing area reported by router = $236,430 \mu^2$

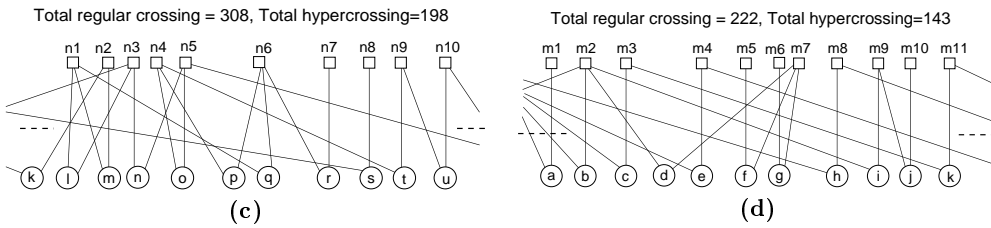


Fig. 1. Layouts and corresponding graph for netlist instance `ref_multi1_16-76-0001`: (a) a section of a layout corresponding to `tr20`, (b) a section of a layout corresponding to `tr19`, (c) graph of the layout in (a), (d) graph of the layout in (b).

(after routing) are reported. The objective of the second tool is to place a netlist of standard cells into variable width rows such that the rows are placed in the *topological order* while cells in each row are permuted to minimize the total wire crossing: the total number of cell levels and the total wire crossing are reported. After a number of repeated experiments with up to 800 netlists in the same equivalence class¹, a definite trend emerges: total wire length as reported by a standard-cell placement tool and the total wire crossing as reported by a graph plotting tool are highly correlated, with correlation coefficients ranging from 0.7 to 0.9 for a number of netlist equivalence classes.

We conjecture that such correlations are not a coincidence, though the cases of placements with multi-row netlists are presently too complicated to analyze. In this paper, we repeat similar experiments in a conceptually simpler context: cell placement in a linear arrangement, and modeling of wires in the channel with a bipartite graph (*bigraph*) $G = (V_0, V_1, E)$.

In our model, V_0 is the set of *net nodes*, V_1 is the set of *cell nodes*, and $E \subseteq V_0 \times V_1$ is the set of edges or wires. A *placement* f maps $V_0 \cup V_1$ to points on the plane so that the y -coordinate of $f(v)$ is i when $v \in V_i$. The cost of *wire crossing*, or the crossing number in the bigraph, is the number of line intersections induced by f . Similarly, a bigraph model is used to formulate a linear arrangement problem and the associated *wire length* cost function [26]. The work in [26] also brings out important relationships between the wire crossing and the linear arrangement problem in graphs. Note that while the wire length

cost depends on the (x, y) coordinates of the nodes, the cost of wire crossing depends only on the *order* of nodes at $y = 0$ and $y = 1$.

In addition, we introduce the concept of *hypercrossing* as follows. Let $G_h = (V_1, H)$ be a hypergraph model of G : include a hyperedge for each net node $v \in V_0$ to connect all cell nodes adjacent to v in G . The edge crossings for a placement f of G induce hyperedge crossings in G_h as follows: hyperedges p and q in G_h have a *hypercrossing* iff there is at least one crossing in G among the edges incident to v_p and v_q , the net nodes corresponding to p and q . Experiments in this paper demonstrate that the sum of all hypercrossings in G has a near-linear relationship to the total wire length as *measured* after rectilinear routing of any given placement.

Our experimental model for the linear arrangement problem is very realistic. We use a state-of-the-art standard cell place and route tool [24] that optimizes a placement of all cell nodes in a single row and then completes the routing in a single channel. The routing is accomplished with vertical and horizontal wire segments in two metal layers, connected through vias. The tool reports the total length of all wire segments in the channel as the *wire length*. An example of routing channel sections for two such placements is shown in Figure 1(a-b). Both sections of the routing channel correspond to placement of 11 2-input NAND cells at the bottom of each channel; cells are omitted for clarity. The corresponding bigraph models corresponding to placements in each routing channel are shown in Figure 1(c-d).

The layouts and graphs in Figure 1 correspond to 11 cell sections from a 67-cell, 66-net netlist example `ref16_multi1_16`. The same netlist is given to two distinct placement tools. Both placements are routed by the *same channel router*. Cells in

¹We leverage the notion of a netlist equivalence class introduced earlier [25]. Netlists in such class have a number of common characteristics such as the same number of nets, cells, pins, logic levels, as well as the *distribution* of cell types at each level (e.g. at level i all circuits would have p_i 2-pin nodes, q_i 3-pin nodes, r_i 4-pin nodes, etc).

the larger layout with 8 tracks have been placed by a standard cell place and route tool that minimizes the wire length; the corresponding regular wire crossing number in the bigraph model is 308 while the total hypercrossing is 198, the routed wire length is $16,726\mu$, and the routing area is $338,670\mu^2$. Cells in the smaller layout with 5 tracks have been placed by a tool that minimizes the wire crossing; the corresponding regular wire crossing number in the bigraph model is 222 while the total hypercrossing is 143 (a 27.8% reduction wrt to hypercrossing of 198), the routed wire length is $13,648\mu$ (a 18.4% reduction), and the routing area is $236,430\mu^2$ (a 30.2% reduction). Given the significant differences in the two layouts, the question arises: *Can we expect to improve the performance of the placement algorithms by minimizing the cost of wire crossing rather than any of the cost models of total wire length in use currently?*

In the next section, we introduce a methodology on the design of experiments that provides the assurance that the significant improvements as observed for this circuit instance are not an isolated case due to chance.

III. DESIGN OF EXPERIMENTS

Design of Experiments and *Experimental Design* are well-established disciplines in sciences and manufacturing processes; keyword entries to popular search engines on the Web return up to 11,259 and 32,086 hits, respectively. However, these hits do not yet include the performance evaluation of algorithms in CAD.

Two of the fundamental principles of experimental design are *randomization* and *replication*. We adopt these principles for the experimental evaluation of heuristics by (1) creating an *equivalence netlist class*, and (2) repeating the experiments for each member in the class. We illustrate the merits of the approach with a very simple example. Using the 67-cell, 66-net netlist example of `ref16_multi1-16-76` introduced in Figure 1 as a *reference netlist*, we create an *isomorphism class* of 100 netlist instances by simply randomizing the order of the nodes in each netlist instance. Simulating any of these netlist produces identical behavior – as expected. However, placing and routing the 100 netlist with our state-of-the-art standard cell place and route tool [24] induces a near-normal *distribution* of measured wire length:

$$\begin{aligned}
 95\% \text{ confidence interval} &= [25, 176\mu, 26, 848\mu] \\
 \text{mean} &= 26, 012\mu \\
 \text{standard deviation} &= 4, 199\mu \\
 \text{maximum} &= 36, 031\mu \\
 \text{minimum} &= 16, 726\mu \\
 \text{maximum reduction} &= 53.6\%
 \end{aligned} \tag{1}$$

where 95%-confidence interval refers to the confidence interval for the mean of the population and the remaining parameters are statistics based on 100 samples from the population class. The *maximum reduction* of 53.6% refers to reducing the wire length from the maximum of $36,031\mu$ (with an ‘unlucky’ starting order) to the minimum of $16,726\mu$ (with a ‘lucky’ starting order) – *without making a single change to the placement algorithm itself*.

The large variations in (1) make a strong case for formalizing the Design of Experiments (DoE) and thereby the scientific basis for the performance evaluation of algorithms in CAD. There is no merit in evaluating and comparing performance of two algorithms on the basis of a single instance of a reference circuit (and report ‘improvements’ even as dramatic as 53.6%). Making comparisons across a few unrelated reference circuits has no

statistical significance either. Variations such as observed and reported for the example of an isomorphism class in (1), are not unusual – they have been reported for a number of netlists and placement algorithms as early as 1986 [27]. Additional and more recent instances are reported elsewhere [25]. Not all instances of reference circuits produce an isomorphism class that induces as much variability as we report in (1). Study of equivalence classes other than isomorphism is required to gain additional insights to this problem. A brief introduction to such classes is given in the next section. For the context of this paper, the basic abstractions for DoE include:

1. an equivalence class of experimental subjects, eligible for a treatment;
2. application of a specific treatment;
3. statistical evaluation of treatment effectiveness.

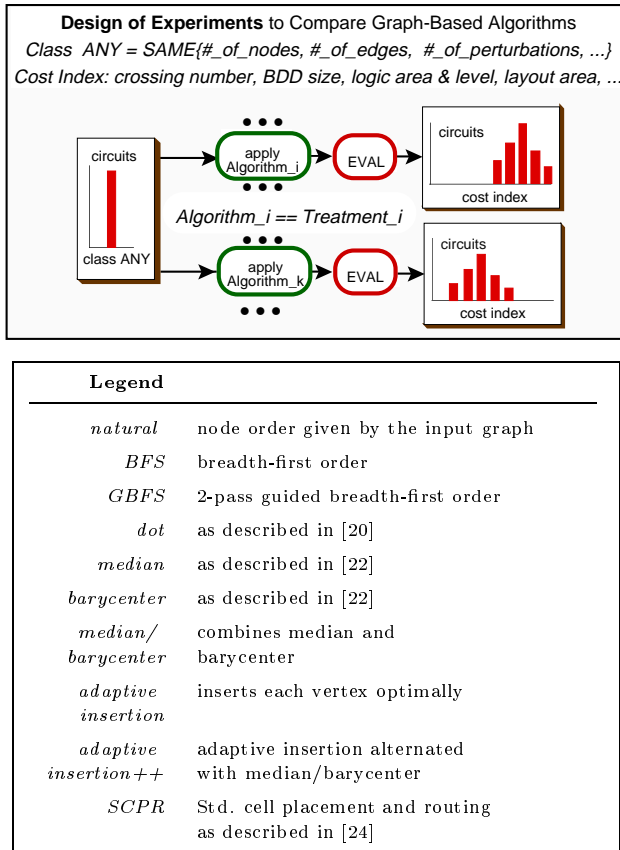
Here, a *treatment* is synonymous with a *heuristic* and an equivalence class of experimental subjects is synonymous with a netlist equivalence class. Figure 2 illustrates these abstractions in a generic flow. In the context of this paper, the cost indexes refer to the cost function minimized by any given placement algorithm: either a model of total wire length or a model of total wire crossing. The companion table shows a list of 21 placements (treatments) for which we have designed experiments reported in this paper: starting with a random placement (tr0) to placements using a drawing package that optimize the cost wire crossing and graph appearance (tr12, tr13 in [20]); placements based on a number of new experimental algorithms that minimize the cost wire crossing (tr14 –tr19 in [23]); and a state-of-the-art standard cell placement and routing tool (tr20 in [24]).

For example, a distribution of wire length induced by Treatment 3 for the same 100 instances of the isomorphism class based on `ref16_multi1-16-76`, the same class used in (1), generates the following statistics:

$$\begin{aligned}
 95\% \text{ confidence interval} &= [22, 630\mu, 23, 922\mu] \\
 \text{mean} &= 23, 276\mu \\
 \text{standard deviation} &= 3, 244\mu \\
 \text{maximum} &= 31, 630\mu \\
 \text{minimum} &= 17, 322\mu \\
 \text{maximum reduction} &= 45.2\%
 \end{aligned} \tag{2}$$

where the parameter description is the same as already provided for (1). Notably, the wire length statistics for the two treatments, tr20 in (1) and tr03 in (2) appear distinct even without a *t*-test [28]. Indeed, for a two-tailed test at the $\alpha = 0.05$ significance level we have $t_{0.975} = \pm 1.972$ while *t*-test based on statistics in (1) and (2) shows that $t = 28.38$, clearly outside the range of $t_{0.975}$. Hence, we can be better than 95% confident that for the isomorphism class based on a `ref16_multi1-16-76`, Treatment 3 results in less wire length on the average than Treatment 20, i.e., it is better by design rather than due to chance.

For each treatment, we measure and report *all three* parameters of interest: total wire length (as reported by the router after placement and routing), total (regular) crossing in the bigraph model, and total hypercrossing in the bigraph model. The simplified flow of the complete experimental design is shown in Figure 3(a). Notably, the wire length is measured in microns in the routed channel of the layout while the respective crossing numbers are technology independent and are subject to relative positions of cell and net nodes after placement only. The example in Figure 3(b), based on data for 100 instances of the isomorphism class of `ref16_multi1-16-76`, demonstrates the correlations between the wire length and hypercrossing distributions for two treatments:



| Treatment | Initial Placement | Final Placement |
|-----------|-------------------|----------------------|
| 0 | natural | none |
| 1 | natural | median |
| 2 | natural | median/barycenter |
| 3 | natural | barycenter |
| 4 | natural | adaptive insertion |
| 5 | natural | adaptive insertion++ |
| 6 | BFS | none |
| 7 | BFS | median |
| 8 | BFS | median/barycenter |
| 9 | BFS | barycenter |
| 10 | BFS | adaptive insertion |
| 11 | BFS | adaptive insertion++ |
| 12 | dot | dot (24 iterations) |
| 13 | dot | dot (48 iterations) |
| 14 | GBFS | none |
| 15 | GBFS | median |
| 16 | GBFS | median/barycenter |
| 17 | GBFS | barycenter |
| 18 | GBFS | adaptive insertion |
| 19 | GBFS | adaptive insertion++ |
| 20 | SCPR | SCPR |

Fig. 2. Design of experiments to compare the placement produced by different algorithms (treatments).

- the wire length distribution as summarized in (1), induced by Treatment 20;
- the hypercrossing distribution induced by Treatment 20;
- the wire length distribution summarized in (2), induced by Treatment 3;
- the hypercrossing distribution induced by Treatment 3.

The positive linear correlation of total wire length with total hypercrossing, and the coefficient of correlation of 98.25% in Figure 3(b) is independent of the treatment we apply. The questions addressed in the remainder of this paper include: (1) what correlations can we expect with equivalence classes other than the isomorphism class, and (2) how will correlation scale as we vary the topology and increase the size of the reference circuits for each class.

IV. EQUIVALENCE CLASSES

The concept of the isomorphism class, introduced informally in the previous section, demonstrated merits and principles for the design of experiments. In this section, we formalize the isomorphism class such that it leads naturally to the more generalized concept of the *mutation class* for bipartite graphs.

Let $G_r = (V_0, V_1, E)$ designate the *reference bipartite graph*. A *characteristic signature* can be any number of mappings based on parameters that relate to the size and distribution of vertices and edges in G_r . For example, to support the definition of a mutation class later in this section, we introduce a signature σ_{mut} :

$$\sigma_{\text{mut}}(G_r) = \{|V_0|, (\text{dist}|V_1|), (\text{dist}(E))\} \quad (3)$$

where $(\text{dist}|V_1|)$ denotes the distribution of vertices in V_1 , classified in terms of edges incident at each vertex, and $(\text{dist}(E))$ denotes the distribution of edges in E classified in terms of *connected components* that constitute G_r .

We create a *perturbation graph* G'_r from G_r as follows: (1) we take a copy of G_r and randomly select and remove q edges from E and put them into an empty urn, (2) we pick edges from the urn and randomly connect vertices V_0 and V_1 , until the urn is empty.

We consider each placement of $G = (V_0, V_1, E)$ as a *presentation* $\langle G, \pi_0, \pi_1 \rangle$, where π_i is a permutation of V_i . Two equivalence classes we use in our experiments are the isomorphism class \mathcal{G}_{iso} and the mutation class \mathcal{G}_{mut} defined as

$$\mathcal{G}_{\text{iso}} = \{\text{presentation of } \langle G_r, \pi_0, \pi_1 \rangle\} \quad (4)$$

$$\mathcal{G}_{\text{mut}} = \{\text{presentation of } \langle G'_r, \pi_0, \pi_1 \rangle | \sigma_{\text{mut}}(G_r)\} \quad (5)$$

where π_i are strictly *random permutations*.

Remarks. A presentation of the perturbation graph, $\langle G'_r, \pi_0, \pi_1 \rangle$, is not necessarily in the mutation class. As expressed in (5), in order to belong to the mutation class, each perturbation graph must also satisfy the signature σ_{mut} in (3) – a non-trivial procedure which has been generalized for arbitrary netlists and is described in [25]. Both the isomorphism class \mathcal{G}_{iso} and the mutation class \mathcal{G}_{mut} satisfy the signature σ_{mut} in (3): both have the same number of nodes, edges, and connected components – as well as the *distribution* of cell types, e.g. all nodes in V_1 have p_i 2-pin nodes, q_i 3-pin nodes, r_i 4-pin nodes, etc. The only random variable in \mathcal{G}_{mut} (relative to \mathcal{G}_{iso}) is number

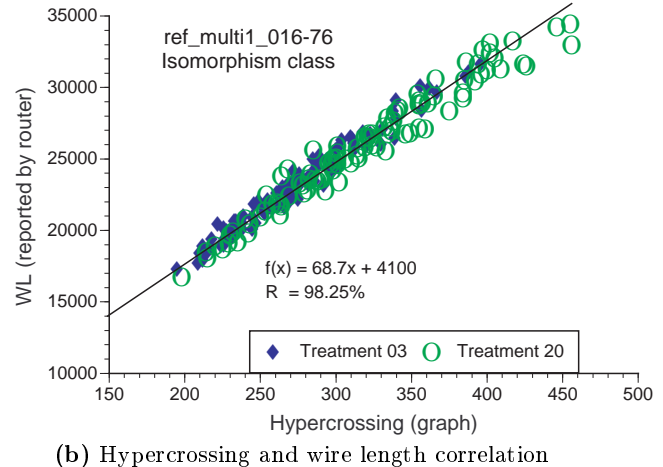
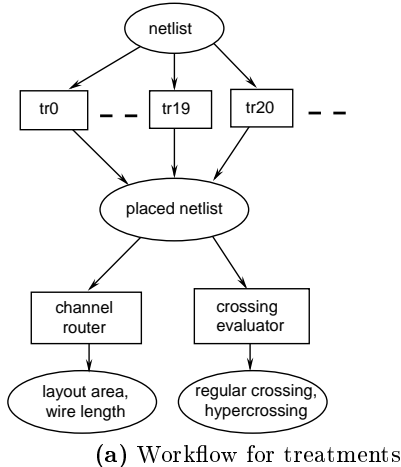


Fig. 3. Workflow for treatments and typical correlations.

of edges assigned to nodes in V_0 , i.e. the fanout of net nodes, bounded by the signature. Satisfying the signature also implies that no graph instance, drawn at random from the equivalence class, will ever have any isolated nodes.

Confounding Properties of Each Class. For all of the experiments reported and the datasets we archive, we took a special precaution to assign to each instance of any class the following properties:

P1: the order of all I/Os and interior nodes in the netlist is random relative to all other members in the class (that is, π_0 and π_1 are uniformly random),

P2: the names of all I/Os and interior nodes in the netlist are assigned randomly relative to all other members in the class.

Properties **P1** and **P2** are essential to good experimental design and must be maintained universally for all equivalence classes. The purpose of **P1** is clear. Without **P2**, some programs that rely on hashing the input data may *unknowingly undo* the randomization of input presentations and confound the experiments. An important lesson on this subject has been learned and reported elsewhere [29].

A number of experiments, reported in the next section, demonstrate the utility and application of both the isomorphism and the mutation classes.

V. EXPERIMENTAL RESULTS

The experiments are based on the model shown in Figure 2, and more specifically, in Figure 3(a). Different placements, i.e., Treatments 0–20, are applied to both the isomorphism and mutation classes of data described in the paper. Unless stated otherwise, each equivalence class consists of 100 instances of netlists with the confounding properties of **P1** and **P2** as defined earlier. All treatments are applied to each instance of a netlist in the class. The netlists of 2-layer graphs in our experiments are representative of connected components extracted from netlists available in [30]. Only seldom would such components have a very large number of nodes – most have but a few nodes, particularly in arithmetic circuits. We find it important to conduct the first set of experiments with cells of the same size; we assume *two input pins* for each cell in the following (sparse) reference graphs:

- **ref_multi1_002**, a 11-cell, 10-net single connected component (32 pins)
- **ref_multi1_004**, a 19-cell, 18-net single connected component (56 pins)
- **ref_multi1_008**, a 35-cell, 34-net single connected component (104 pins)
- **ref_multi1_016**, a 67-cell, 66-net single connected component (200 pins)
- **ref_multi1_032**, a 131-cell, 130-net single connected component (392 pins)
- **ref_multi1_008Xn**, a concatenation of $n = 1, 2, 3, 4$ **ref_multi1_008** graphs (representing $n = 1, 2, 3, 4$ connected components with a total of 104, 208, 312, 416 pins) to be used in the second part of the experiments in evaluation of the asymptotic behavior of placement algorithms.

For each presentation of input $\langle G, \pi_0, \pi_1 \rangle$, a treatment h_k induces a treated presentation $h_k(\langle G, \pi_0, \pi_1 \rangle)$ as $\langle G, \pi'_0, \pi'_1 \rangle$ which is in turn evaluated by *common evaluator tools* that process uniformly the results of all treatments:

- technology-specific **channel_router** that completes all connections in the channel and reports *layout area and total wire length* of all rectilinear connections in the channel;
- technology-independent **crossing_evaluator** which analyzes the placement reports *regular crossing and hypercrossing*.

All classes of input data are being organized for easy access through the Web; web-posting will include complete tables, for each treatment and each instance of a netlist from an equivalence class, parameters such as layout area, wire length, (regular) wire crossing, and hypercrossing. Statistical summaries that include confidence intervals of the reported means, and *t*-tests to analyze the significance of reported differences between the means condense the results of these tables. In this paper, Treatment 19 is found to be the best overall treatment, and we also post cell permutations π'_0, π'_1 along with the respective routed wire length and crossing numbers reported by this treatment. Reader of this text should look under http://www.cbl.ncsu.edu/experiments/-DoE_Archives/DoE_0004 for completed archives of experimental data presented in this paper.

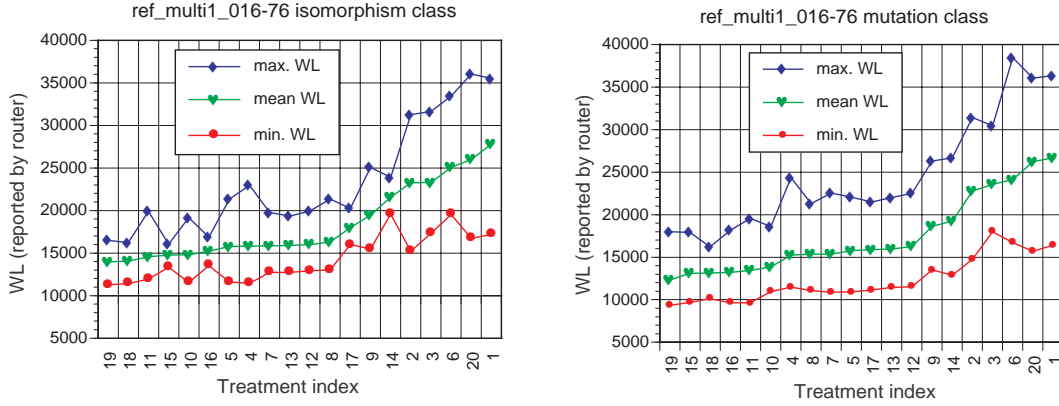


Fig. 4. Wire length in layout, as reported by the router, for Treatments 0-20 for two equivalence classes.

Due to space limitations, we confine the statistical summaries of the experiments to composite illustrations shown in Figures 4, 5, and 6. We discuss our observations in the order of these illustrations next.

Figure 4. The experiments summarized in Figure 4 are designed to reveal:

- variability and performance of 20 linear placement algorithms (Treatments 1 – 20) in contrast to random placement (Treatment 0, representing the case of *no treatment* or a *placebo treatment*);
- the sensitivity of each treatment relative to the two input data classes: the isomorphism class and the mutant class;
- the relative ranking of all treatments with respect to total wire length for both equivalence classes.

Data, as arranged in Figure 4 (sorted on the average total wire length for each treatment), lead to following observations and tentative conclusions:

- Regardless of the equivalence class, Treatment 19 consistently outperforms all other treatments. The variability of wire length distribution is greater for the mutation class but not dramatically.
- Regardless of the equivalence class, Treatment 20 (based on direct minimization of wire length) is consistently a poor performer relative to all other treatments except Treatment 1. The variability of wire length distribution is greater for the mutation class but not dramatically.
- All other treatments are ranked between Treatment 19 and Treatment 20; there is some but not dramatic changing of ranks when treatments are applied to the isomorphism class and the mutation class. However, Treatments 14, 15, 16, and 17 are displaying significantly less variability in distributions for the isomorphism class relative to the mutation class.
- All treatments significantly outrank random placement (Treatment 0). The variability of wire length distribution is greater for the mutation class but not dramatically.
- The isomorphic class in Figure 4 is generated using a specific mutant (#76) from the mutant class as a ‘reference’ and represents one of the ‘harder’ isomorphic classes we observed – i.e. ones where variability of reported wire length is as variable as the one shown for this isomorphism class. Clarifying what ‘makes an isomorphic class’ more difficult from another, is subject to research and conjectures.

Overall, Figure 4 provides a template for results of a useful experiment that should be repeated for (a) additional classes of circuits, (b) at least one more state-of-the-art placement and routing tool driven by the objective of wire length minimization.

This objective is being pursued currently and additional results will be reported later.

Figure 5. The experiments summarized in Figure 5 are based on the template of the correlation experiment introduced in Figure 3(b). The main objectives of these experiments are:

- to evaluate the correlation, for a given placement, between the total hypercrossing in the graph model and the total wire length, after rectilinear routing of all connection in a single channel.
- to assess the variability of correlation between the total hypercrossing and the total wire length in response to different treatments.
- to assess the variability of correlation between the total hypercrossing and the total wire length in response to different mutation classes.
- to examine the correlation between the (regular) wire crossing in the graph and the hypercrossing.

Data, as arranged in Figure 5, lead to following observations and tentative conclusions:

- The mutant class `ref_multi1_016` (a 67-cell, 66-net single connected component with 200 pins) is large enough to visually differentiate between the performance of Treatment 19 and Treatment 20. However, both treatments induce the correlation coefficient of 99.5%, comparable to 98.3% induced by Treatments 3 and 20 in Figure 3(b) on the isomorphism class `ref_multi1_016`.
- As expected, the correlation of hypercrossing to regular crossing for the mutant class `ref_multi1_016` is not as good. The companion correlation graph shows a correlation coefficient of 88.6%.
- The mutant class `ref_multi1_032` (a 131-cell, 130-net single connected component with 392 pins) shows significant difference between the performance of Treatment 19 and Treatment 20. Here, two very distinct treatments induce the correlation coefficient of 99.7%!
- In the companion graph, we demonstrate that the correlation coefficient is truly independent of the treatment. Here, we superimpose Treatment 20 and Treatment 0 (random placement). Notably, both the slope and the correlation are close to those for Treatment 19 and Treatment 20. Furthermore, Treatment 20 does represent a significant improvement over Treatment 0. Overall, Figure 5 provides a template to present results of any number of correlation experiments. Additional experiment will be conducted for additional classes of circuits.

Figure 6. The experiments summarized in Figure 6 are designed to measure:

- asymptotic performance of *two* placement algorithms (Treatments 19 and 20) and *two* equivalence classes – *all while the*

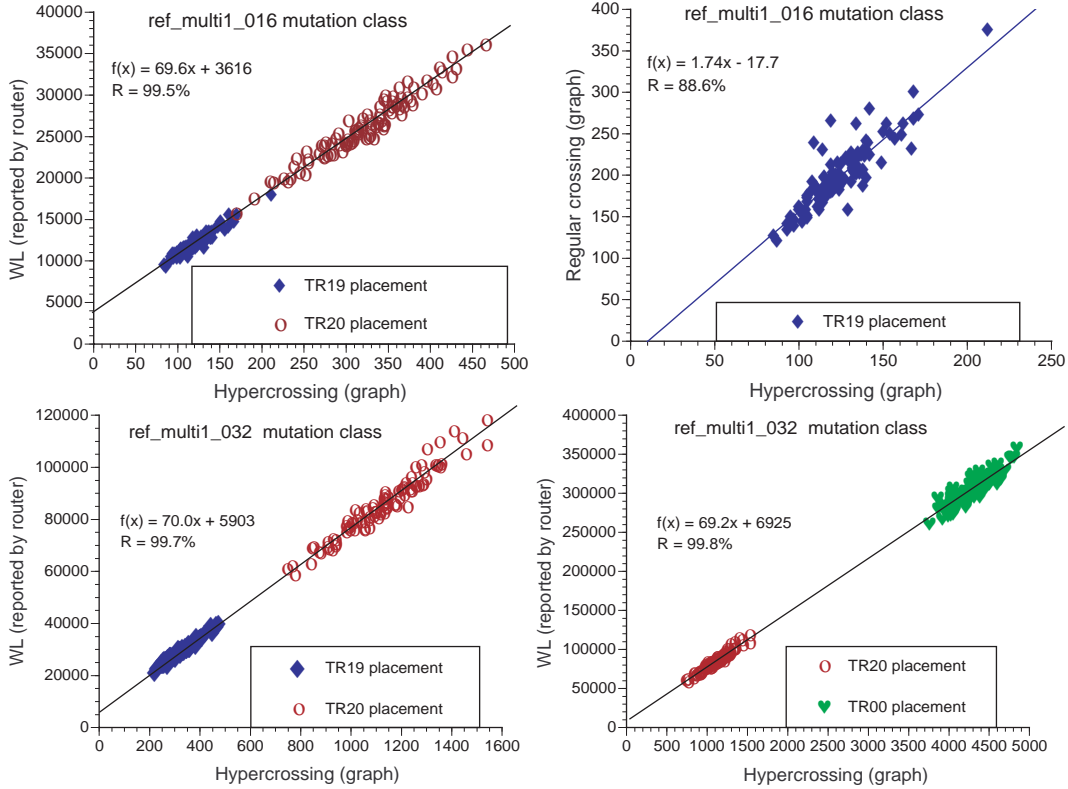


Fig. 5. Wire length in layouts versus wire hypercrossing and regular crossing in graphs.

netlist doubles in the count of the pins and each instance of the graph remains as a single connected component. We create the families of isomorphism and mutation classes based on `ref_multi1_00n` ($n = 2, 4, 8, 16, 32$) for this purpose (32, 56, 104, 200, 392 pins).

- asymptotic performance of *two* placement algorithms (Treatments 19 and 20) and *two* equivalence classes – *all while the netlist increases linearly in the count of the pins and each instance of the graph becomes either 1, 2, 3, or 4 connected components, each of the same size.* We create the families of isomorphism and mutation classes based on `ref_multi1_008xn` ($n = 1, 2, 3, 4$) for this purpose (104, 208, 312, 416 pins).

Data, as arranged in Figure 6, leads to following observations and tentative conclusions:

- First, consider the isomorphism and mutation classes of `ref_multi1_00n` ($n = 2, 4, 8, 16, 32$). Here, the performance of Treatment 19 for the isomorphism class is comparable to the performance of Treatment 19 for the mutation class. Similarly, the performance of Treatment 20 for the isomorphism class is comparable to the performance of Treatment 20 for the mutation class. Note however, that the gap between the performance of Treatment 20 versus Treatment 19 is growing rapidly: from about $1,200\mu$ at 200 pins to $6,000\mu$ as the number of pins increases to 392. There is no doubt that Treatment 19 will continue to perform significantly and increasingly better than Treatment 20 as the size of graph instances increases in either class.

- Next, consider the isomorphism and mutation classes of `ref_multi1_008xn` ($n = 1, 2, 3, 4$). Here, *only the performance of Treatment 19 for the isomorphism class is comparable to the performance of Treatment 19 for the mutation class.* In contrast, the gap in performance of Treatment 20 versus Treatment 19 is about $12,000\mu$ in the isomorphism class at the point of 416

pins. However, the gap in performance of Treatment 20 for the mutation class at the point of 416 pins is wider still: about $85,000\mu$!! We attribute the widening gap to the fact that, unlike Treatment 19, Treatment 20 is clearly not aware of the four isolated connected component construction of this equivalence class when it reaches the size of 416 pins. Moreover, problem instances for the Treatment 20 appear much harder with the mutation class than the isomorphism class. Note however, there is no such difference for Treatment 19; for both equivalence classes, the wire length increases linearly with the increasing count of the pins in the graph – as we would expect from an algorithm that finds connected components first, given the construction of these graphs.

VI. CONCLUSION AND FUTURE WORK

This paper is only the beginning of what we hope is a new approach to experimental study of the placement problem. We could not have performed as detailed correlation and comparison studies without using the isomorphism and mutation equivalence classes. Minimization of total wire hypercrossing rather than minimization of total wire length may lead to a new generation of placement algorithms.

Using the Web, input data and treated output can be shared and verified so that different groups working on the same problem can conduct repeatable experiments. Better heuristics are often developed through a detailed understanding of why specific instances present difficulties, and such understanding is made more likely when various data equivalence classes are significant component of the experimental design. Some directions to pursue later include:

- Experiments with alternative linear placement algorithms that are minimizing the cost of total wire length.
- Comprehensive experiments with multi-row placement algo-

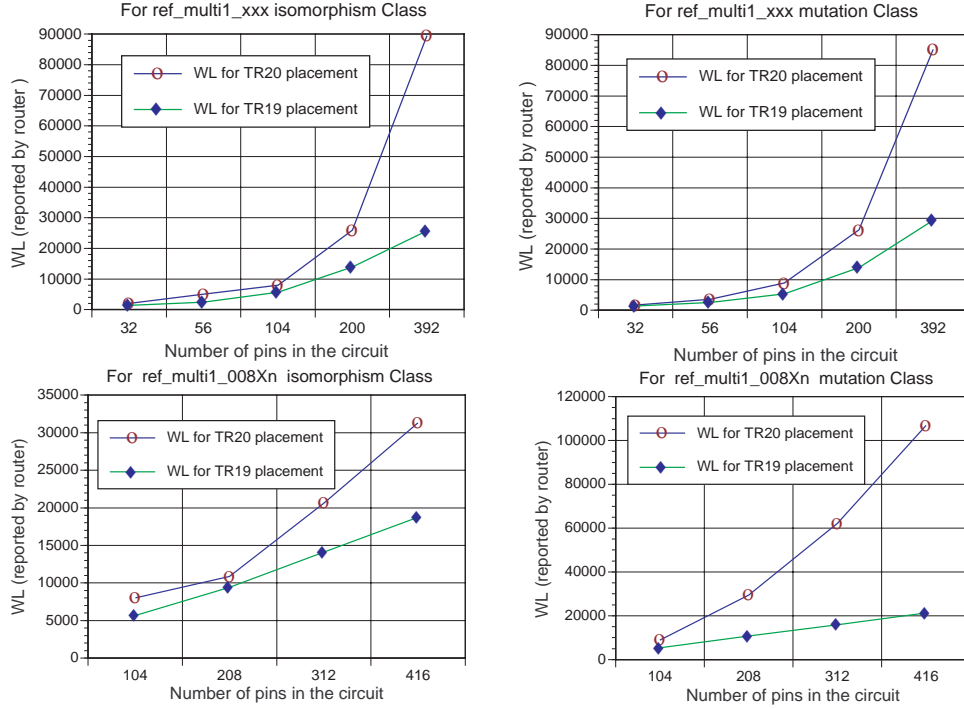


Fig. 6. Asymptotic behavior of treatments 19 and 20.

rithms, with a number of equivalence classes as inputs, with correlation studies of routed wire length versus hypercrossing similar to ones presented in this paper.

- New placement heuristics that directly minimize the cost of hypercrossing (rather than regular crossing).
- A generalized model that supports correlation of hypercrossing to wire length with modules that may be of different size and shape.

For an up-to-date summary of all data presented in this paper, see [31].

VII. APPENDIX

In this section we report additional results of experiments regarding the correlations among the three parameters of interest: wire length in layout, regular wire crossing seen in a biplanar embedding of the graph and the hypercrossing in the graph. We also report results with additional equivalence classes, different from those discussed in Section V. We also show a comparison of asymptotic behavior of the treatments 19 and 20 for these equivalence classes.

Figure 7. This is a detailed correlation study for the three parameters - wire length, regular crossing and hypercrossing for the same two equivalence classes of Figure 5. It is evident that although there exists reasonable correlation between wire length and regular wire crossing, it is the hypercrossing that correlates better than 99% with the wire length. Hence hypercrossing, and not the regular wire crossing, should be the topological parameter to deal with for optimizing the geometrical parameters such as layout wire length. Presently, we do not have an algorithm that minimizes hypercrossing. On the other hand, there are good heuristics for regular wire crossing minimization, Treatment 19 being one of the best known to date [23]. Hence, we attempt to minimize the hypercrossing indirectly by minimizing the regular crossing. There is a good correlation between regular crossing and hypercrossing, as seen in Figure 7. Thus, minimizing regular crossing is a good heuristic for minimizing

hypercrossing, and in turn, the layout wire length.

Figure 8. We wanted to check whether the above observations hold good for two-layer graph equivalence classes other than the `ref_multi1` series, especially for two-layer graphs seen in typical circuits. Hence, we picked one level from the circuit `classifier` or `cc` from [30] (all the single-input nodes have been removed from this graph since they do not add any complexity in terms of wire crossing). The resulting graph is shown in Figure 8. We replicated this graph 1, 2, 4 or 8 times to form 4 reference graphs `cc_lev1Xn`, $n = 1, 2, 3, 4$. Both isomorphism and mutation classes were generated for these 4 reference graphs and the layout experiments repeated with treatments 19 and 20.

Figure 9. Here, we report the correlations among wire length, regular wire crossing and hypercrossing for the mutation classes `cc_lev1X4` and `cc_lev1X8`. The observations are similar to results in Figure 7, further supporting the fact that wire length follows hypercrossing more closely than the regular crossing, although regular crossing minimization can be used to minimize hypercrossing indirectly.

Figure 10. We report the correlations among wire length, regular wire crossing and hypercrossing for the isomorphism classes `cc_lev1X4` and `cc_lev1X8`. Notably, Treatment 19 always returns the same value for the cost function (as one expects with an optimal algorithm and an isomorphism class). However, Treatment 20, failing to take advantage of isolated connected components in the graph, produces results with wide variance.

Figure 11. Here, we report the asymptotic performance of treatments 19 and 20 as we increase the problem size. The observations of Figure 6 have been repeated for the equivalence classes `cc_lev1Xn`, ($n = 1, 2, 3, 4$) which are very different from the equivalence classes `ref_multi1_00n` ($n = 2, 4, 8, 16, 32$) or `ref_multi1_008Xn` ($n = 1, 2, 3, 4$) of Figure 6. Treatments 19 and 20 are quite comparable for `cc_lev1X1`, but differ widely (the performance ratio being 7:1 for the isomorphism class and 5:1 for the mutation class) for the largest graphs.

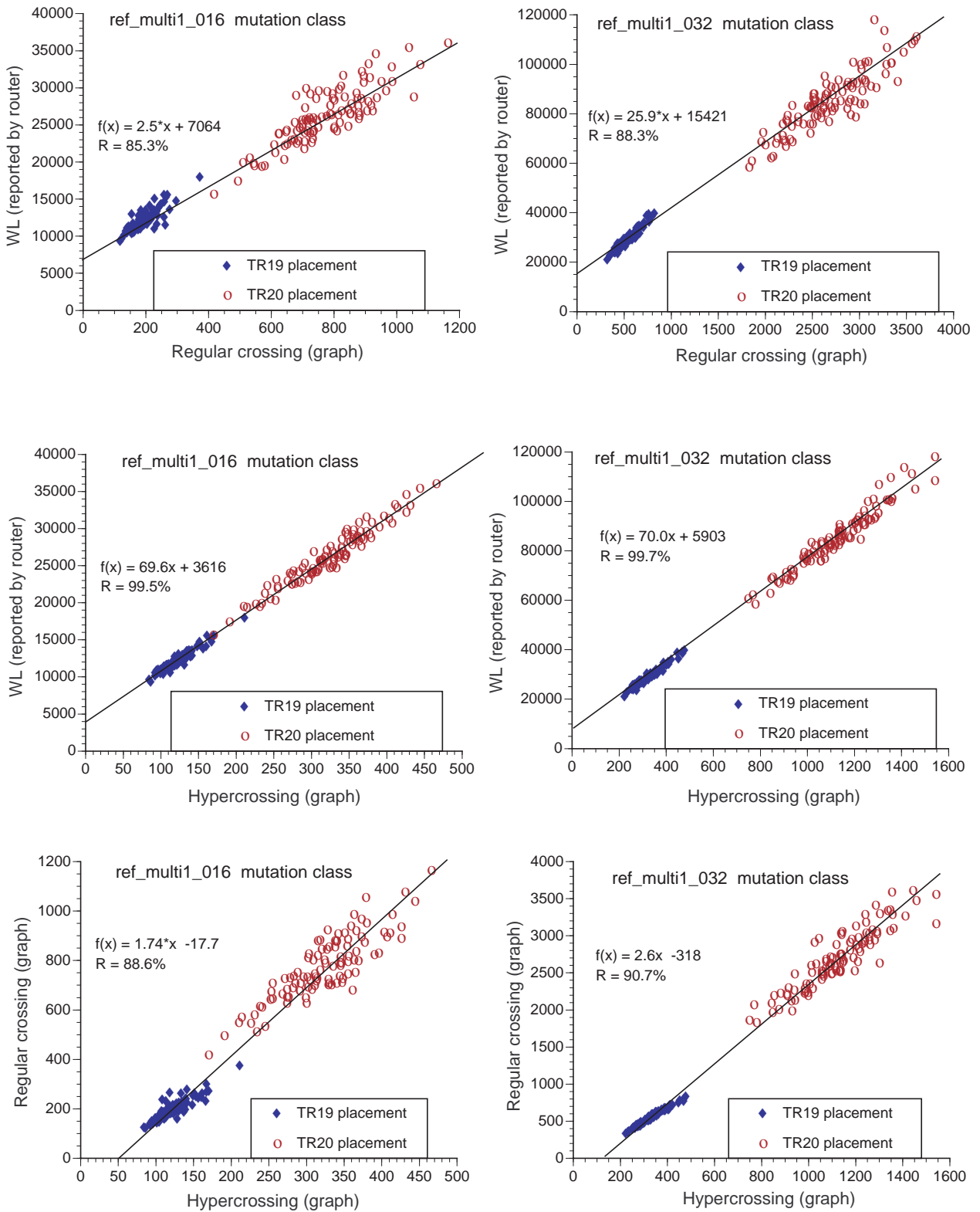


Fig. 7. Correlations between wire length in layout and regular crossing, wire length and hypercrossing and regular crossing and hypercrossing.

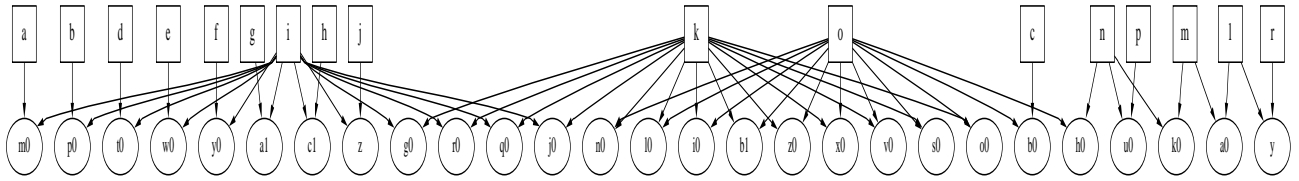


Fig. 8. First level of the MCNC benchmark `cc.blif` that is used for generating the graph equivalence classes `cc_lev1Xn`.

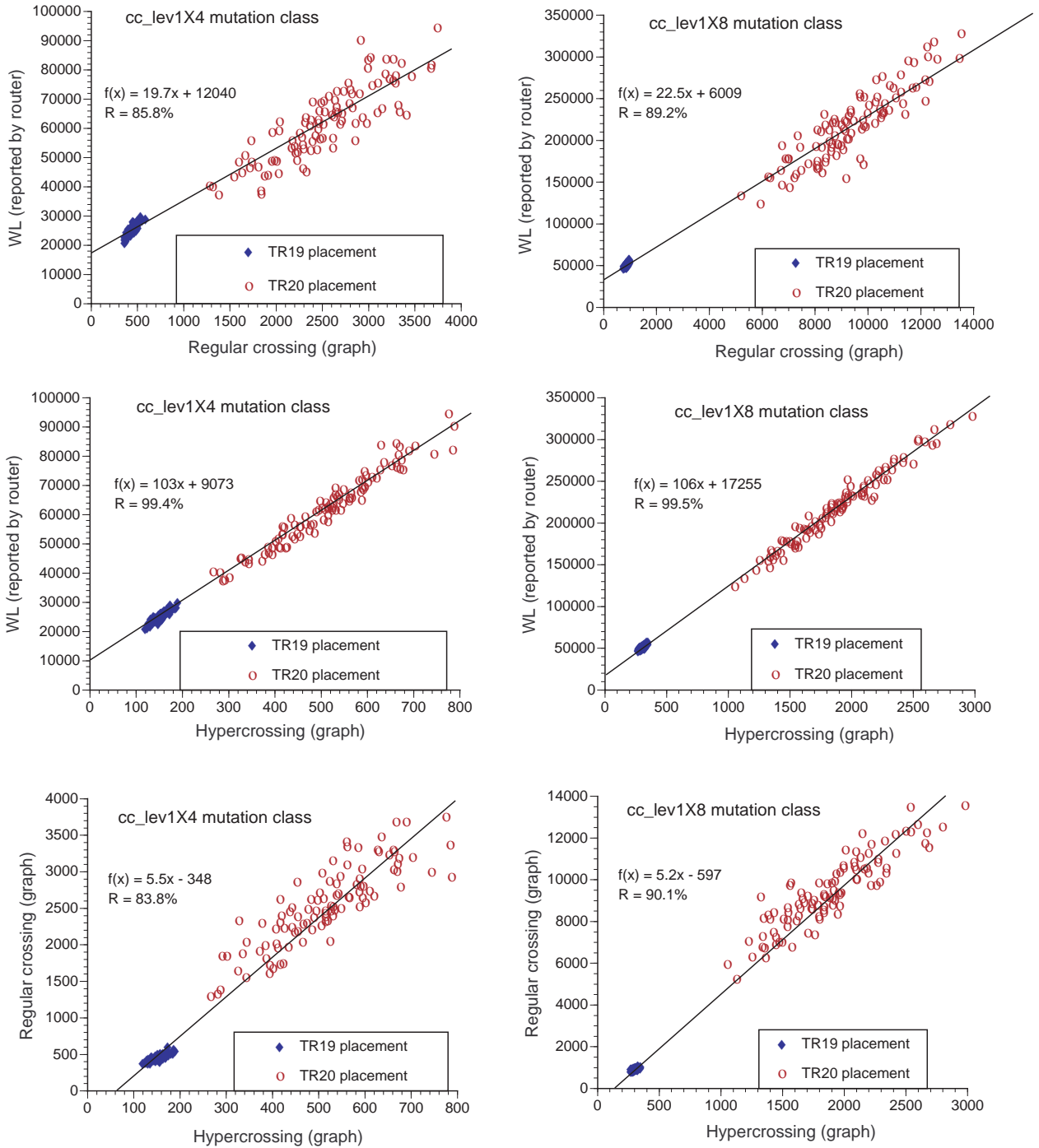


Fig. 9. Wire length in layout versus hypercrossing and regular crossing and correlation between hypercrossing and regular crossing.

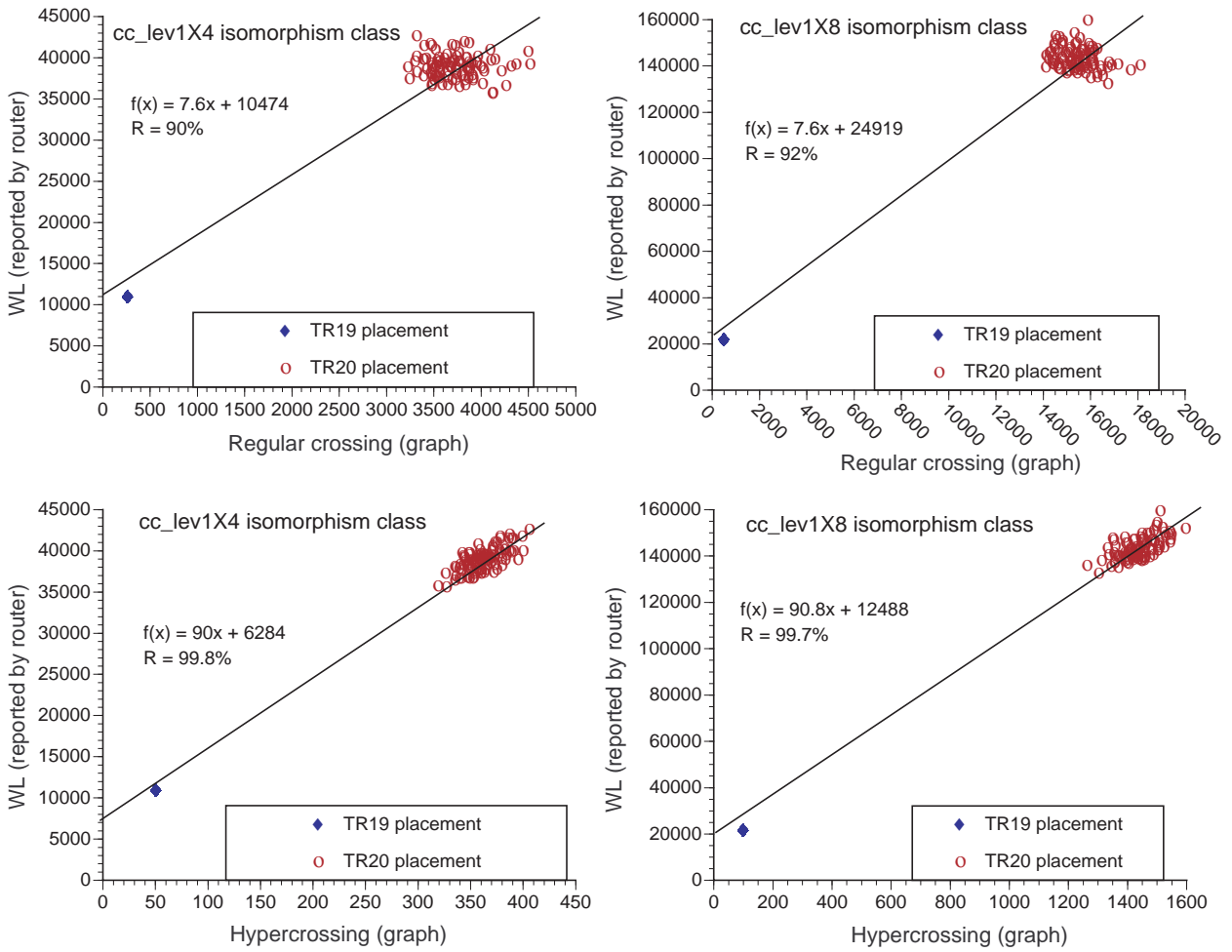


Fig. 10. Wire length in layout versus hypercrossing and regular crossing.

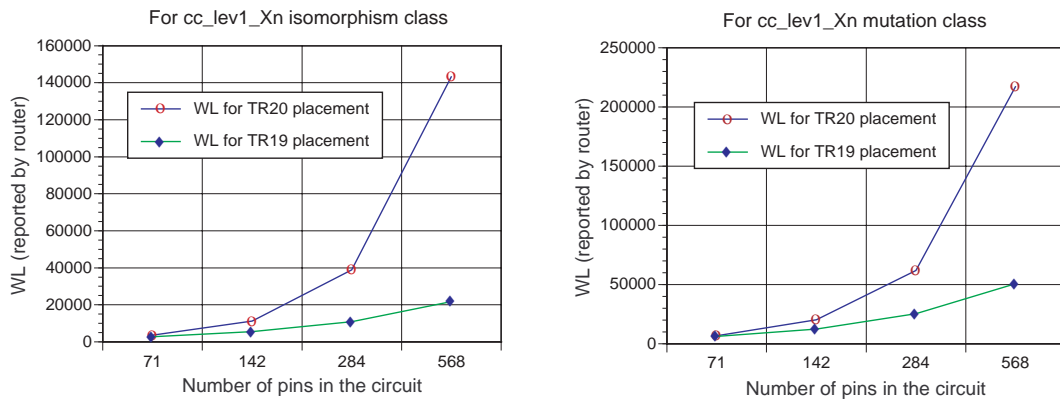


Fig. 11. Asymptotic behavior of treatments 19 and 20 for `cc_lev1_Xn` equivalence classes.

ACKNOWLEDGMENTS. Some of the initial crossing number experiments with 2-layer graphs posted on the Web in 1997 were organized and executed by Nevin Kapur. He also contributed the software utilities that support gathering and tabulating data about the experiments and the statistical summaries. Hemang Lavana contributed utilities to dynamically generate web-pages of directories that link the data related to the reported experiments. We appreciate their support.

REFERENCES

- [1] C. Sechen. *VLSI placement and global routing using simulated annealing*. Boston: Kluwer Academic, 1988.
- [2] W. Sun and C. Sechen. Efficient and Effective Placement for Very Large Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(3):349–359, March 1995.
- [3] P. R. Suaris and G. Kedem. A quadrisection-based combined place and route scheme for standard cells. *IEEE Transactions on Computer-Aided Design*, 8(3):234–244, 1989.
- [4] K. Doll, F. M. Johannes, and G. Sigl. Accurate net models for placement improvements by network flow methods. In *Proc. Int. Conf. on Computer-Aided Design*, pages 594–597, 1992.
- [5] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich. GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization. *IEEE Transactions on Computer-Aided Design*, 10(3):356–365, March 1991.
- [6] G. Sigl, K. Doll, and F. Johannes. Analytical placement: A linear or quadratic objective function? In *Proc. 28th ACM/IEEE Design Automation Conference*, pages 427–432, 1991.
- [7] A. Srinivasan, K. Chaudhary, and E. S. Kuh. RITUAL: A performance driven placement algorithm for small cell IC's. In *Proc. Int. Conf. on Computer-Aided Design*, pages 48–51, 1991.
- [8] R.S. Tsay and E. Kuh. A unified approach to partitioning and placement. *IEEE Transactions on Circuits and Systems*, 38(5):521–533, 1991.
- [9] C.K. Cheung and E.S. Kuh. Module placement based on resistive network optimization. *IEEE Transactions on Computer-Aided Design*, CAD-3:218–225, 1984.
- [10] C. J. Alpert, T. F. Chan, D. J. Huang, A. B. Kahng, I. L. Markov, P. Mulet, and K. Yan. Faster minimization of linear wirelength for global placement. In *Proc. 1997 International Symposium on Physical Design*, pages 4–11, April 1997.
- [11] I.I. Mahmoud, K. Asakura, T. Nishibu, and T. Ohtsuki. Experimental appraisal of linear and quadratic objective functions effect on force directed method for analog placement. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E77-A(4):719–725, April 1994.
- [12] B. M. Reiss, K. Doll, and F. M. Johannes. Partitioning very large circuits using analytical placement techniques. In *Proc. 31st ACM/IEEE Design Automation Conference*, pages 646–651, 1994.
- [13] C. Sechen. Average Interconnection Length Estimation for Random and Optimized Placements. In *IEEE International Conference on Computer-Aided Design*, pages 190–193, November 1987.
- [14] T. N. Bui and S. Lee. On the Mincut Bipartite Arrangement Problem. In *IEEE International Conference on Computer-Aided Design*, pages 466–469, November 1987.
- [15] F. T. Leighton. New Lower Bound Techniques for VLSI. *Math. Systems Theory*, 17:47–70, 1984.
- [16] M. Marek-Sadowska and M. Sarrafzadeh. The Crossing Distribution Problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(4):423–433, April 1995.
- [17] H.F.S. Chen and D.T. Lee. On Crossing Minimization Problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(5):406–418, May 1998.
- [18] J. N. Warfield. Crossing Theory and Hierarchy Mapping. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(7):505–523, July 1977.
- [19] M. R. Garey and D. S. Johnson. Crossing Number is NP-complete. *SIAM J. Algebraic Discrete Methods*, 4:312–316, 1983.
- [20] E.R. Gansner, E. Koutsifios, S.C. North and K.P. Vo. A Technique for Drawing Directed Graphs. *IEEE Trans. Software Engg.*, 19:214–230, 1993. The drawing package `dot` is available from <http://www.research.att.com/sw/tools/graphviz/>.
- [21] The AGD-Library: Algorithms for Graph Drawing, 1998. Available from <http://www.mpi-sb.mpg.de/mutzel/dfgdraw/agdlib.html>.
- [22] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, 1999.
- [23] M. Stallmann, F. Brglez, and D. Ghosh. Heuristics and Experimental Design for Bigraph Crossing Number Minimization. In *ALENEX'99, the First Workshop on Algorithm Engineering and Experimentation*, January 1999. A preliminary version is available as a technical report from <http://www.cbl.ncsu.edu/publications/#1999-ALENEX-Stallmann>.
- [24] (Ed.) K. Kozminski. OASIS 2.0 User's Guide. MCNC, Research Triangle Park, N.C. 27709, 1992. (Over 600 pages, distributed to over 60 teaching and research universities worldwide).
- [25] D. Ghosh, N. Kapur, J. E. Harlow, and F. Brglez. Synthesis of Wiring Signature-Invariant Equivalence Class Circuit Mutants and Applications to Benchmarking. In *Proceedings, Design Automation and Test in Europe*, pages 656–663, Feb 1998. Also available at <http://www.cbl.ncsu.edu/publications/-#1998-DATE-Ghosh>.
- [26] F. Shahrokhi, O. Sýkora, L. A. Székely, and I. Vrt'o. On bipartite drawings and the linear arrangement problem. In *Lecture Notes in Computer Science*, number 1467, pages 55–68. Springer Verlag, 1997.
- [27] M. R. Hartoog. Analysis of Placement Procedures for VLSI Standard Cell Layout. In *23rd Design Automation Conference, ACM/IEEE*, pages 314–319, July 1986.
- [28] K. A. Brownlee. *Statistical Theory and Methodology In Science and Engineering*. Krieger Publishing, 1984. Reprinted, with revisions, from second edition, 1965.
- [29] J. E. Harlow and F. Brglez. Design of Experiments in BDD Variable Ordering: Lessons Learned. In *Proceedings of the International Conference on Computer Aided Design*. ACM, November 1998. Also available from <http://www.cbl.ncsu.edu/publications/#1998-ICCAD-Harlow>.
- [30] S. Yang. Logic Synthesis and Optimization Benchmarks User Guide. Technical report, MCNC, Research Triangle Park, NC, 1991. Now available from <http://www.cbl.ncsu.edu/publications/#1991-IWLS-UG-Saeyang> and benchmarks from <http://www.cbl.ncsu.edu/benchmarks/Benchmarks-upto-1996.html>.
- [31] Archival directory for the experimental design DoE_0004, 1998. Accessible at http://www.cbl.ncsu.edu/experiments/DoE-Archives/DoE_0004.