[22] G. Yee and C. Sechen, "Clock-delayed domino for adder and combinational logic design," in *Proc. IEEE/ACM Int. Conf. Computer Design*, Oct. 1996, pp. 332–337.

[23] J. K. Zhao, E. M. Rudnick, and J. H. Patel, "Static logic implication with application to redundancy identification," in *Proc. IEEE VLSI Test Symp.*, Apr. 1997, pp. 288–293.

---

# Reporting of Standard Cell Placement Results

Patrick H. Madden

***Abstract*—Very large scale integration (VLSI) fabrication technology has advanced rapidly, bringing with it a strong demand for faster and better design automation tools. Accurate reporting of results for placement approaches is crucial to the development of improved automation tools; unfortunately, publicly available placement benchmarks are outdated, and there are wide variations in their interpretation. In addition, the metrics considered by some academic research have questionable relevance to modern design.**

**At best, poor benchmarks and differences in interpretation result in misunderstandings of the effectiveness of some approaches. At worst, they can motivate research in areas of very little promise, while other areas which have true potential are ignored. In this paper, we expand on work previously presented, describing current standard cell placement benchmarks and illustrating common differences in their interpretation. We also propose specific interpretation methods for traditional objectives, and discuss new metrics which should be considered in modern placement research. Our hope is that by presenting these issues clearly, we can enable more accurate evaluations of placement methods, and improve research efficiency.**

***Index Terms*—Benchmarks, metrics, placement, standard cell design.**

## I. INTRODUCTION

Standard cell placement is a fundamental problem in very large scale integration (VLSI) computer-aided design (CAD). Objectives such as wire length and area minimization have long been a concern, and with the advent of deep submicron design, the scope of the problem now includes delay optimization, power minimization, and a number of other issues.

Many approaches to the placement problem have been proposed, and a set of well-known benchmark circuits is widely available. Unfortunately, optimization objectives commonly considered by academic research groups are outdated: comparisons based on these objectives are of questionable value. To compound this problem, there is wide variation in benchmark interpretation, making fair comparisons on even simple metrics difficult or impossible.

In this paper, we attempt to classify the common interpretations, and bring together reported results from a number of authors into a unified table. The large variation in results make it clear that there are fundamental differences in how measurements are taken. Contrary to what might be anticipated for a problem this well studied, there are few "common assumptions." By making these issues clear, we hope to facilitate the more accurate comparison of methods. We also discuss modern concerns, which are not necessarily captured by the traditional objectives of wire length and area minimization.

This paper has a motivation similar to that of [2], which illustrated difficulties in hypergraph partitioning research, due to variation in implementation of "standard" algorithms. The challenges faced with circuit placement are substantially more difficult than those observed with hypergraph partitioning: there is not even agreement on how a result should be measured, to say nothing of how a "standard" algorithm should be implemented. While there has been a convergence of results in hypergraph partitioning, with many algorithms matching the best observed performance [1], it is unclear what sort of result a "good" placement method should achieve.

As with [2], we observe that the lack of consensus on interpretations hampers placement research. In many published works, there are unintentional comparisons of "apples" to "oranges," resulting in incorrect or misleading conclusions. If we hope to promote the best algorithms and approaches, we must have a clear understanding of what "best" means. At the most fundamental level, if we hope to derive any benefit from research results, we must know what the results imply.

The remainder of the paper is organized as follows. We first describe modern concerns, which are not necessarily those considered in early placement research. We next provide a bit of historical perspective on the benchmarks themselves, followed by a discussion of the metrics used to evaluate placement approaches. We then present a table which brings together reported results from a number of earlier works; a wide difference in reported results makes it clear that there is considerable range in how benchmarks are interpreted. Our next section presents a suggested approach to simplifying benchmark interpretation, and also provides new placement results (which are available in their entirety through the web) for use in future comparisons. We conclude this paper with a number of suggestions that should provide for less ambiguity in the evaluation of placement methods, allowing research to be directed more efficiently and effectively.

## II. MODERN CONCERNS

Wire length minimization and area minimization are "traditional" optimization objectives, but these are not only interesting design goals. It might be argued that these are actually of secondary concern; if a circuit operates with acceptable power consumption levels and clock rates, a circuit designer would have only a passing interest in metrics such as wire length or circuit area.

We stress this issue, as much of the remainder of this paper focuses on wire length, and we do not wish to suggest that minimized wire length is synonymous with quality. In this section, we discuss briefly a number of "modern concerns," which we classify as geometric and electrical.

### A. Geometric Concerns

A major modern concern is the routability of a placement, which cannot be predicted by wire length and area alone. After placement, transistors must be connected by wiring, and this wiring requires physical space. This problem is "geometric" in nature, and like many other physical design problems (for example, Steiner tree construction and block packing), is intractable.

If the circuit is to be fabricated in a fixed amount of area, a placement with low wire length may still encounter a *congestion* problem, where we are unable to successfully route the design. Relatively little academic work considers placement results *after* successful routing, but this is clearly a more interesting problem.

---

Manuscript received April 20, 2001; revised September 17, 2001. This work was supported in part by the National Science Foundation under Grant CCR-9988222. This paper was recommended by Associate Editor M. D. F. Wong.

The author is with the Department of Computer Science, State University of New York at Binghamton, Binghamton, NY 13902 USA (e-mail: pmadden@cs.binghamton.edu).

Publisher Item Identifier S 0278-0070(02)01053-9.

While area minimization is a traditional objective in academic placement work, it has become less critical recently. Many modern fabrication processes utilize a *fixed die* model; we are faced with a hard upper limit on the area available to lay out the circuit, but there may be no benefit for using less area. This contrasts with the earlier *variable die* model, in which it was desirable to minimize circuit area as much as possible.

### *B. Electrical Concerns*

With the continual scaling of feature size, physical design problems now also have important aspects that can only be captured with sophisticated electrical models. Power consumption, system delay, and signal integrity, for example, cannot be measured with simple geometric rules, and evaluation of "quality" is nontrivial. This complicates the evaluation of placement approaches considerably, and we will refer to these concerns as performance issues.

A fundamental concern that is difficult to capture effectively is coupling capacitance, in which adjacent routing wires form capacitors. Coupling capacitance impacts system delay significantly, can substantially increase power consumption, and can even cause voltage fluctuations which may result in incorrect values being latched into registers. Wire adjacencies are only known after both global and detail routing, so this is impossible to predict accurately at the placement stage.

Meeting performance objectives is perhaps the most important concern for circuit designers. If a circuit fails to obtain the necessary clock rates, or consumes too much power, it will be commercially impractical, and the designers effort will be for naught. The actual performance can only be known at a very late stage in the physical design flow, and frequently requires knowledge of device parameters that are proprietary information of the foundry. While optimization for these concerns is clearly desirable, technical and legal obstacles make it difficult for academic groups to pursue this effectively.

## III. STANDARD CELL PLACEMENT BENCHMARKS

The MCNC benchmark suite was released in the early 1990s, with a number of standard cell circuits being made available in the *YAL* and *VPNR* formats. Several translators were available, allowing conversion into EDIF, and academic TimberWolf formats. Later, other translators allowed conversion into a variety of formats, including PROUD, commercial versions of TimberWolf, Cadence LEF/DEF, and the Gigascale Silicon Research Center (GSRC) Bookshelf formats. While not part of this original group of benchmarks, IBM's *golem3* has also become a staple in placement research. These circuits are by far the most commonly used benchmarks in standard cell placement research.

In [16], results of early placement approaches on the benchmark circuits were summarized. In this work, only *circuit areas* were reported; the circuits were placed and routed, with the figure of merit being total area. While it was suggested that subsequent research should report circuit areas (and in particular, for placements with a variety of aspect ratios), most work reports instead *half perimeter wire lengths*.

The transition from reporting of area to reporting of half perimeter wire length is understandable. Global and detail routing are generally time consuming, and variation in the quality of the routing tools can skew comparisons. Routing is now rarely performed (for reporting of placement results), and half perimeter wire length is widely accepted as a relevant metric. While one might wish for more accurate metrics, achieving this has been difficult.

The MCNC benchmarks were designed utilizing fabrication parameters that were current at the time. The advance of fabrication has made many of the fundamental assumptions used in these designs inappropriate for modern design.

1) In early fabrication processes, only two metal layers were available for routing, requiring additional "channel" space between active circuit elements. With current processes, more metal layers are available, and most routing can occur "over the cell."
2) For the early benchmarks, a simple $RC$ delay model is suggested. This is clearly inadequate; most groups now use either *Elmore* [7] delay, or an approach based on asymptotic waveform evaluation [18].
3) The benchmarks are small compared to current circuits, with the larger MCNC benchmarks (*avqsmall, avqlarge*) being half the size of IP blocks expected for system-on-a-chip design.
4) Modern issues such as power minimization, crosstalk minimization, and signal integrity, are not considered at all.

Industry groups have been reluctant to release new benchmarks, with competitive advantage being a primary concern. As a result, researchers in academia are left with a dilemma. If the MCNC placement benchmarks are used as designed, relevance of results to modern objectives is questionable. If proprietary (industry) benchmark results are reported, other groups will be unable to perform comparable experiments; there is no way of knowing if reported results are good, bad, or indifferent. If the MCNC placement benchmarks are scaled or adapted to modern fabrication technologies, comparison of results to previous work will be hopelessly skewed.

## IV. REPORTING OF RESULTS

In this section, we illustrate common benchmark interpretations, which are a large factor in the variation in reported results. In some cases, we have been unable to determine what interpretations were used to produce some reported results. The details we are concerned with here are rarely documented in the published work, and due to time constraints, we were unable to contact all authors directly. We consider first those issues which impact wire length estimates, and then address performance optimization concerns.

### *A. Wire Length Metrics*

While wire length might seem to be a relatively simple quantity to measure, there are a surprising number of issues where results can be skewed substantially.

*1) Differences in Translation:* With each translation, there is the possibility that information can be lost. In most cases, the numbers of cells and nets are preserved. The naming scheme, however, is not always preserved, making determination of the use of some nets unclear. Errors in translation may be subtle, and are, therefore, difficult to identify. In some reported work, we observe the following.

a) The number of cells or nets in a benchmark varies from the specifications of the original circuits. This clearly indicates that the results cannot be compared to previously published work.
b) Names of nets and cells are sometimes lost in translation. Clearly, nets named *Vdd, Vss, Reset, Clock, Scan, Phi1,* or *Phi2,* should receive different treatment than ordinary nets, and may be critical to proper determination of circuit delay.

*2) Scaling of Benchmark Dimensions:* To enable fast cell mirroring, early versions of the TimberWolf placement tool required cell dimensions to be even integers. To handle this requirement, the MCNC benchmarks *fract*, *struct*, and *biomed* have their dimensions *doubled* in the TimberWolf formats, while *golem3* has dimensions multiplied by four.

The result of this scaling is that in some cases, results reported for these benchmarks may differ by a factor of two or four depending on what input files are used. This has clearly occurred: in [20], the result for *golem3* is reported as 88.98, while [26] reports a result of 19.84, and scales the [20] result to 22.60.

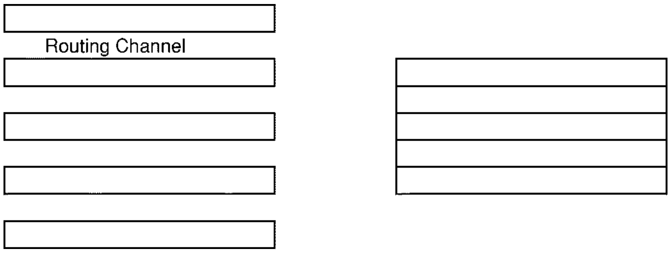

Fig. 1. In early fabrication processes, additional space between cell rows is required to complete routing. With modern fabrication, over the cell routing allows the elimination of space between cell rows in most cases. We can expect significantly reduced wire lengths for modern fabrication processes, even if we account for feature size scaling.

*3) Row Length and Spacing, and Numbers of Rows:* Perhaps the most significant difference in interpretation has been with row spacings and the numbers of rows used in standard cell placement. When the MCNC benchmarks were first released, relatively few metal layers were available for routing; thus, *channel based design* is appropriate, and spacing between cell rows is required. With current fabrication techniques, *over the cell* routing is possible, and additional spacing can be eliminated in most cases.

Changes to row spacing result in a substantial impact to estimated placement wire lengths. In Fig. 1, a simple five row placement is shown; by removing spacing between rows, we reduce the lengths of nets which span more than one row. Both interpretations of row spacing are reasonable, and both occur within the literature. In our experiments, we observe that it is quite possible to have as much as a 30% reduction in wire lengths simply by removing inter-row spacing.

If row spacings are not determined precisely by complete routing, many groups assume spacing is equal to standard cell height. In [14], [20], and [13], results reported are with routing between cell rows. In [6], results are obtained with row spacing equal to cell height. In [11], [26], and [12], no space is assumed between rows.

With variable die placement, minimization of row length is a common concern. Increase in the length of the longest row increases the die size, and thus the fabrication cost. In some early placement tools, the maximum row length is restricted severely, with variations in length being quite small (TimberWolf [21], for example, commonly obtains less than 1% variation). In fixed die placement, this constraint is relaxed substantially, and it is possible to obtain substantially reduced wire lengths in this situation.

The number of rows used has also varied widely. This has substantial impact on results. If we consider a circuit which forms an 8 by 8 mesh, a placement into 8 rows is clearly superior to either a 7 or 9 row solution. There is no way to *a priori* determine an appropriate number of rows, or to determine the impact of this decision on total wire length. For the benchmark *primary2*, the number of rows used have included 29 [16], 36 [15], 28 [20], [6], [8], [12], [26], and 32 [19]. The benchmark *primary2* is not unusual; there are a variety of row numbers used for the other benchmarks as well.

*4) Pad Positioning:* A benchmark feature frequently obscured in translations between formats is the placement of input and output pads. In some cases, the pads are brought near to the cell rows; in others, positions are determined by the row spacings expected (using routing channels).

As with row spacing, changes to pad positions can impact the reported wire length, particularly for small benchmarks where a large percentage of nets connect to pads. These differences are illustrated in Fig. 2.

*5) Pin Positions:* Half perimeter wire length is the most common length-based metric used in research on standard cell placement. Like issues already mentioned, there are a number of reasonable methods to estimate half perimeter wire length, and instances of many of these are in use.

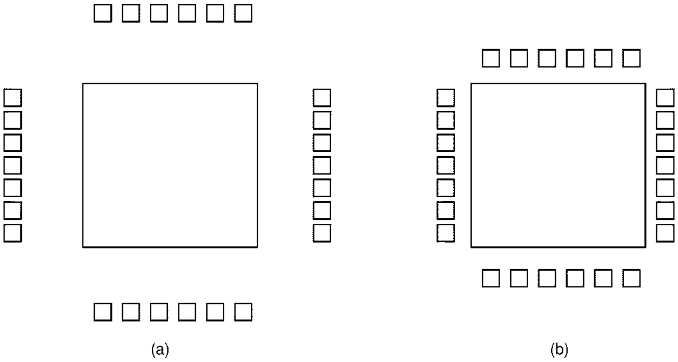

Fig. 2. Pad positions are not always preserved during translation from one format to another. If pads are shifted away from standard cell rows, this obviously increases expected wire length. If we assume modern fabrication techniques, placing the pads at the positions suggested in the original VPNR and YAL benchmark files would be extremely pessimistic.

Pin positioning is more complex than it might initially appear. In most standard cell libraries, there are multiple *port* locations to allow connection to the transistor inputs and outputs. These ports may be on either the tops or bottoms of the cells, or follow a "center terminal" alignment. Some current detail routers remove predetermined connection wiring, placing vias at any convenient and design-rule correct location.

The variation in the number of ports, and their locations, results in variations in half perimeter wire length estimation methods. In Fig. 3, a number of reasonable methods are shown.

In Fig. 3(a), the bounding box contains the pins of a net in their entirety; this is perhaps the most conservative estimate. In Fig. 3(b), the bounding box is determined by the centers of the cells, and ignores precise pin positions; this is a frequently used metric, which eliminates the impact of cell mirroring. In Fig. 3(c), the bounding box contains pins that might be included with a Steiner or Spanning Tree construction. In Fig. 3(d), the bounding box contains the lower left corner of each cell, producing an estimate that would be similar to that of Fig. 3(b), but with some differences depending on cell placements and sizes. Some placement tools use the first pin defined in the cell library to determine pin locations, resulting in the possibility that a number of equivalent placements could have slightly different wire length estimations.

Again, these metrics are all reasonable, but can result in differing wire length estimates. The metric shown in Fig. 3(b) might be most common, but the estimate of Fig. 3(c) might be more accurate.

*6) Spanning and Steiner Tree Metrics:* In some work ([11], for example), wire length estimates are based on Spanning or Steiner Tree constructions, resulting in an estimate that would be more accurate than half-perimeter. Half-perimeter estimates could be considered "optimistic" for nets with more than three pins. If all other factors are equal, one would expect the highest wire lengths to be reported for metrics based on spanning trees, with Steiner tree metrics reducing lengths slightly, and half perimeter resulting in the lowest length estimates.

### B. Other Metrics

While most of our focus is on issues that impact half-perimeter wire length measurement, we note that there is considerable interest in performance optimization (in terms of delay, power consumption, and signal integrity), as well as "practical" metrics such as run time and memory requirements.

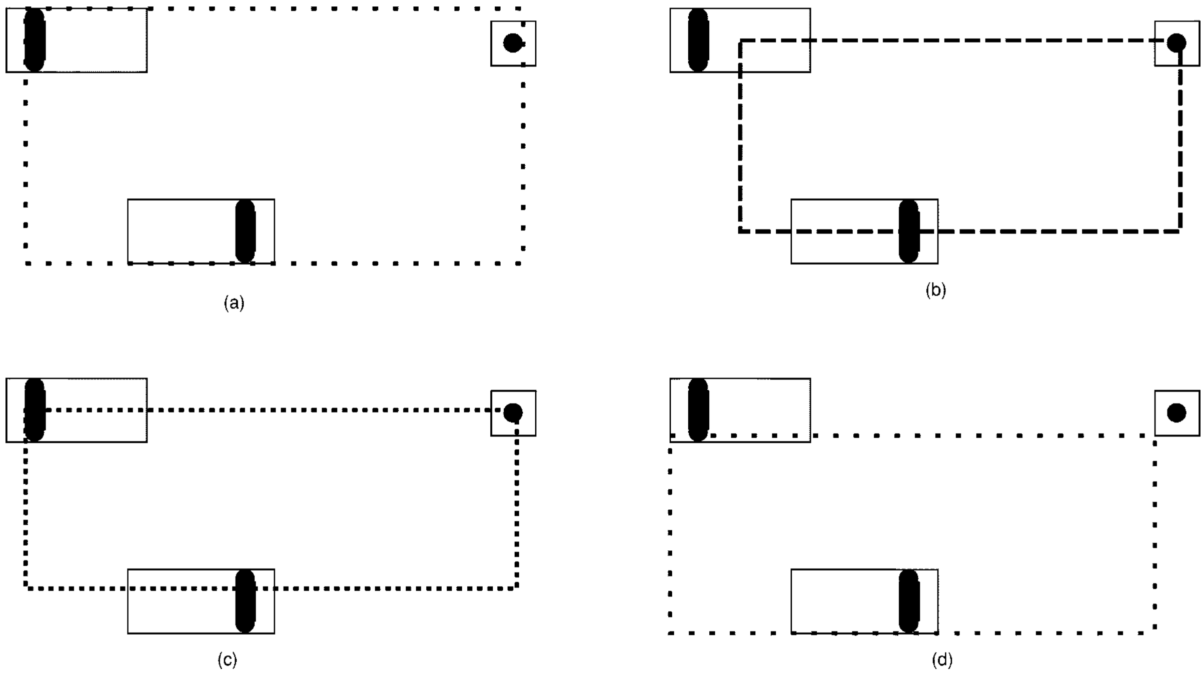

Fig. 3. Estimation of half perimeter wire lengths can be done in a number of ways. In most cell libraries, there are multiple ports for the inputs and outputs of a cell; how these locations are considered, or if we only consider cell locations and ignore actual pins, impacts the computed wire length.

*1) Delay, Power, Signal Integrity:* We address in more detail here, some issues mentioned in Section II. The initial MCNC circuits were designed for the technology parameters of the day. We focus in this section on delay optimization, but other modern concerns face similar challenges. Obviously, current performance driven research must consider current fabrication technology, resulting in difficulty in the comparison of results. If current work utilizes the earlier technology parameters, it has little relevance to modern design. If the work instead considers modern parameters, we can expect that wire lengths and delays would be substantially reduced. We note again that this issue has serious implications.

a) If the original device dimensions are used, wire length results are comparable, but delay analysis will consider wire lengths greatly in excess of what could be reasonably expected.
b) If device dimensions are scaled to modern technology, comparison to previous work is impossible.

As interconnect delay continues to increase in importance [4], these scaling issues become progressively more difficult. If we are to obtain reasonable and realistic evaluations of timing-driven placement approaches, we must scale feature sizes and transistor sizes carefully. The delay model suggested in [16] is clearly inappropriate for modern design; more complex delay models are required. In this area, it is likely that interpretation decisions will have far greater impact than any algorithmic choice we could make.

Beyond these scaling issues, the problem of simply determining the longest critical path is quite difficult. Most modern circuitry utilizes sophisticated clocking schemes, and careful architecture design, to minimize the longest path. Simple methods such as determination of the longest path through a circuit may identify false paths (resulting in optimization for an unrealistic objective). If the functionality of individual nets is not considered, optimization may focus on large nets (such as reset nets, or scan chains), missing the nets that are of true consequence.

*2) Routability:* Half perimeter wire length is only an estimate of the routing resources required to complete a design. If the routing is disproportionately horizontal or vertical, or is unevenly distributed, it may be impossible to completely route the circuit. Without successful routing, a placement is of little use [3].

Many modern fabrication processes use *fixed dies,* in which both circuit area, and the spacing between rows, cannot be changed. Most placement research considers only wire length minimization, but it is possible that a solution with low wire length might not "fit" into the space available, or routing of the placement might fail.

### *C. Additional Considerations*

Beyond simply placement quality, we have a number of other issues which can influence the evaluation of results.

*1) Run Times and Random Starts:* Run times are frequently reported for each benchmark, but comparisons may be difficult or misleading. First, comparisons of processing power on different computing platforms is nontrivial. Second, implementation details such as language choice may impact run times, but reveal little about the underlying algorithmic complexity.

To complicate the issue further, many placement tools can take advantage of increased run times to obtain improved results. The impact of multiple random starts on partitioning results is well-known, so placement methods based on partitioning can simply utilize more starts, increasing run time while generally improving solutions. With annealing-based methods, additional run time can be utilized for additional moves at a temperature step, or for an elongated cooling schedule.

Ultimately, there can be tradeoff between run time and solution quality, with a circuit designer perhaps being best suited to determine the right mix for a specific design. In published work, authors usually attempt to provide "reasonable" run time results; but what could be considered "reasonable" changes with the advance of computing platforms.

As many placement algorithms utilize randomization, the results of a single run may vary. To obtain a better measure of average case behavior, multiple runs may be required. The number of runs used when

TABLE I
NUMBER OF CELLS AND NETS IN MCNC PLACEMENT BENCHMARKS. DEPENDING ON THE ROW SPACING AND DESIRED ASPECT RATIO, THE NUMBER OF ROWS USED VARIES

| Benchmark | Cells | Nets |
|---|---|---|
| fract | 149 | 163 |
| struct | 1952 | 1920 |
| primary1 | 833 | 904 |
| primary2 | 3014 | 3029 |
| biomed | 6514 | 7052 |
| industry1 | 3085 | 2594 |
| industry2 | 12637 | 13419 |
| industry3 | 15433 | 21967 |
| avqsmall | 21918 | 30038 |
| avqlarge | 25178 | 33298 |
| golem3 | 100312 | 217362 |

reporting the best observed performance, however, varies; when results are compared, differing numbers of runs, or total run times, may cause the comparison to be unfair.

*2) Tuning:* Many placement tools have parameters which can be "tuned." In bisection-based methods, for example, different cut sequences can lead to substantially different results [25]. With annealing based methods, the type of moves considered may also have considerable impact.

In general, it is difficult to determine optimum parameter values for an individual benchmark, and each benchmark might require distinctly different parameter settings. It is possible that the default configuration of a placement tool will not produce the best possible result for a given benchmark. In practice, it is reasonable to assume that a design team would become familiar with a tool, and be able to make adjustments to match design constraints; for reporting of results, either adjustment or use of default parameters might be considered unfair.

*3) Tool Versions:* In many published works, results from TimberWolf are reported. We make special note of this, as there have been many versions of this placement tool, and the version numbering sequence causes some confusion. TimberWolf transitioned from an academic tool, with the highest version being "7.0," to a commercial tool, with a beginning "1.0" version number. Thus, reported results using "TimberWolf 1.x" are likely references to a fairly recent commercial tool, and not to a very early academic effort. An academic version of TimberWolf is frequently integrated with the Berkeley LAGER package, but this is not the best performing version of the tool, with a default configuration that prefers speed to quality of result.

## V. REPORTED RESULTS

We first summarize the MCNC benchmarks (and golem3) in Table I, reporting the number of cells and nets in each. By modern standards, the benchmarks could be considered small.

We now present Table II, which summarizes wire lengths reported for the MCNC benchmark circuits. Each column corresponds to a tool which has been presented in a competitive conference or journal publication, with the exception of the last column, which was obtained from InternetCAD, a commercial tool vendor. Obviously, this is only a subset of the work done on standard cell placement. We select this set as they provide a cross-section of results, and are all reasonably well-known works.

We have contacted many of the authors cited in this table; uniformly, they have been extremely helpful in clarifying their results. Due to time constraints, we have not been able to contact all authors, and this illustrates a central concern of this paper. Ideally, we would hope that the published record would allow a clear understanding of the results of research; in practice, this is seldom the case.

We report the results *verbatim* from the cited works, and in roughly chronological order. The "benchmark units" used has varied, and we include this detail in the second to last row of the table. Units range from meters to microns, resulting in a difference in where a "decimal point" should be placed. The final row of the table includes an indication of row spacing.

Unfortunately, we cannot provide a clearer picture of how the various placement approaches compare. Normalization of results to a common metric is not possible; changing the number of rows in a placement or the locations of pins within a cell can cause significant changes to wire lengths. Without actually implementing these changes and performing new experiments, we have no way to know how the results of a particular placement tool would be affected.

Given the length of time that the MCNC benchmarks have been considered, and the importance of the problem, one might expect some sort of convergence of the reported results. Clearly, this is not the case; even for the smallest benchmarks, results within the last few years differ substantially.

In the next section, we present new translations of the MCNC benchmarks and an evaluation method which should remove most of the ambiguity in their interpretation. We also provide placements for the circuits which should serve as good "reference points," and briefly describe placement tools which are freely available in both source code and executable form. This should allow independent verification of results, easy construction of true "apples to apples" experiments, and allow a new foundation from which subsequent comparisons can be made.

## VI. IMPROVING BENCHMARKING: THE GSRC BOOKSHELF

Clearly, the current benchmarking situation is not satisfactory. In discussions with a number of researchers, a general consensus has been that an appropriate next step is to develop and promote a common metric; implementation details for the metric are of secondary concern.

In this section, we describe work done in cooperation with Prof. Andrew B. Kahng, who has established the "Bookshelf" [9], a repository for benchmarks, tools, and documentation on a number of VLSI CAD related topics. The bookshelf is only one component of the GSRC, a broad multiuniversity research effort. The bookshelf borrows much from the open source paradigm: contributions are accepted from volunteers, and information is freely and openly exchanged. The goal of the bookshelf is in part to gather and organize information of use to researchers, and to provide archival of research tools and results. The bookshelf contains "slots" for topics ranging from Boolean satisfiability to interconnect synthesis.

The "standard cell placement slot" has received contributions from a number of individuals, particularly Prof. Kahng, A. E. Caldwell, I. L. Markov, X. Yang, and M. C. Yildiz. Our contribution to the slot has been to provide assistance in design of a new benchmark format, and the translation of MCNC benchmark circuits into this format.

### *A. New Translations of MCNC Benchmarks*

While there is an abundance of VLSI file formats, those involved in the standard cell placement slot have chosen to develop a new format to describe benchmark problems. The motivation for a new format comes from a desire to make the format easily interchangeable with other problems (particularly hypergraph partitioning, routing, and interconnect synthesis), and to remove ambiguity in interpretation.

We have developed translations of the MCNC benchmarks (and Golem3) in the bookshelf placement file formats. Examples and detailed descriptions of the formats are available on the bookshelf website. Each file is human-readable, and a complete placement problem is described by files that contain a description of the circuit

TABLE II

REPORTED WIRE LENGTH RESULTS. A SIGNIFICANT DIFFERENCE IN INTERPRETATION IS IN *ROW SPACING*, WHICH VARIES FROM AREA REQUIRED BY AN ACTUAL ROUTING, TO AN ESTIMATE OF CHANNEL HEIGHT EQUAL TO ROW HEIGHT, TO A COMPLETE ELIMINATION OF SPACING

| | [14] Gordian | [10] PRC | [23] HALO | [15] POPINS | [20] TW7.0 | [11] QUAD | [19] NRG |
|---|---|---|---|---|---|---|---|
| Fract | | | | 45225 | | 337 | 25602 |
| Struct | 558+362 | | | | | 3780 | 287631 |
| Primary1 | 841+553 | 999.5 | 1.439 | 1128245 | 0.83 | 8972 | 894545 |
| Primary2 | 4761+3153 | 3665.6 | 6.73 | 4128324 | 3.53 | 36824 | 3412195 |
| Biomed | 5232+3123 | | | 4128324 | 3.22 | 23765 | |
| Industry1 | | | | | | | |
| Industry2 | | | | | 13.30 | 332318 | |
| Industry3 | | | | | 41.53 | 938682 | |
| Avqsmall | | | | 23848188 | 5.08 | 62890 | |
| Avqlarge | | | | 28323022 | 5.65 | 65906 | |
| Golem3 | | | | | 88.98 | | |
| Units | X1000 | X1000 | Meter | Micron | Meter | X100 | Micron |
| Spacing | Routed | | | | Routed | None | |

| | [6] FD98 | [8] ARP | [22] Dragon | [26] SPADE | [12] Mongrel | [24] Feng Shui | [13] iTools |
|---|---|---|---|---|---|---|---|
| Fract | | 0.034 | | 0.024 | | 0.032 | |
| Struct | 0.338 | 0.34 | | 0.291 | 0.266 | 0.380 | 0.272 |
| Primary1 | 0.87 | 0.79 | | 0.74 | 0.83 | 1.018 | 0.799 |
| Primary2 | 3.72 | 3.61 | | 3.13 | 2.94 | 3.684 | 3.37 |
| Biomed | 1.78 | 1.83 | | 1.43 | | 1.689 | 2.90 |
| Industry1 | | 1.50 | | | | 1.606 | |
| Industry2 | 14.6 | | 12.88 | 11.90 | 11.89 | 15.408 | 11.4 |
| Industry3 | 45.1 | 48.12 | 42.33 | 35.37 | 34.53 | 44.729 | 39.6 |
| Avqsmall | 4.91 | 6.06 | 5.17 | 5.08 | 4.4 | 5.960 | 4.48 |
| Avqlarge | 5.38 | 6.54 | 5.25 | 6.16 | 4.87 | 6.301 | 4.78 |
| Golem3 | | | 77.56 | 19.84 | | 21.882 | 79.9 |
| Units | Meter | Meter | Meter | Meter | Meter | Meter | Meter |
| Spacing | Row | | Row | None | None | Row | Routed |

net list, the locations of cell rows, the size and shape of individual cells, and a file which gives exact locations for each cell and pad.

There has been an effort to keep file formats simple and intuitive. Each of the input files can be parsed with extremely simple code; the parser within our *Feng Shui* [24], [25] placement tool, for example, consists of a few loops which use *scanf* to read input lines. Source code for parsing the input is available, and we expect that construction of a parser from scratch would only take a few days. The new format is also supported by *Capo* [3], while *Dragon* [22] and *Mongrel* [12] are completing support. At least one commercial tool vendor is also planning support for the new format.

The design flow possible with the bookshelf formats is far from complete; placement results can be accurately evaluated, however. Prof. Kahng's research group has released a wire length calculation tool that can be used to verify half perimeter wire lengths results (measuring both center-of-cell and cell origin bounding boxes); this tool also checks to ensure that cells are placed within standard cell rows, and are nonoverlapping.

Ideally, subsequent placement research would use these new benchmark files (or be able to produce compatible files through translation). Reported wire lengths could then be easily verified, preventing any accidental confusion or misinterpretation. We also *very strongly* suggest that placement results should be made available through the web; there is little in the MCNC benchmarks that could be considered proprietary information, and the availability of placement results would allow increased confidence in any comparisons made.

### B. Unambiguous Metrics

Interpretation of wire length within the bookshelf placement file formats is precise. Row locations and spacing are *fixed*, as are pad positions. Net wire length is measured from the *centers* of cells, as indicated in Fig. 3(b). Circuit wire length is the sum of the half-perimeter bounding boxes.

If a placement tool uses this formulation, results should be identical to those reported by the wire length calculation tool. If differences exist, it is a clear indication that one of the tools is in error.

Objectives such as congestion, system delay, and routability, are not currently supported; the intention of the bookshelf group is to develop evaluation methods for these issues, and also software tools to verify reported results.

It should be obvious that if we cannot rely on accurate reporting of wire lengths, we can have little confidence in reported "performance driven" placement results. Net delay depends on the resistance and capacitance of interconnect wires, and these, in turn, depend on interconnect length.

### C. New Placement Results for Comparison

In this subsection, we present placement results from *Feng Shui*, using the benchmark input files described. Two versions of the benchmarks are available: one which uses row spacing equal to standard cell height (as is found in much early placement work), and one which uses no row spacing (as is found in some current placement work).

*Feng Shui* is a variable-die placer: it attempts to balance row lengths, approximately minimizing core area. For placement problems in which there is very little excess area or "white space," it may fail to produce a feasible solution. The number of rows used, maximum row lengths, percentage of unused core area (white space), and center-to-center half perimeter wire lengths, are shown in Table III. *Feng Shui* is fundamentally a bisection-based placer, following the general structure proposed by Dunlop and Kernighan [5]; as such, these results can be considered as those of a "classic" method. The source code for *Feng Shui* is available both on our research group web page, and also through the bookshelf site, while command line parameters used are contained within

TABLE III

PLACEMENTS RESULTS FOR *FENG SHUI* USING THE NEW GSRC BOOKSHELF TRANSLATIONS OF THE MCNC BENCHMARKS. TWO CONFIGURATIONS ARE CONSIDERED: ONE IN WHICH ROW SPACINGS THAT ARE EQUAL IN HEIGHT TO A STANDARD CELLS, AND ONE WHICH CONTAINS NO INTER-ROW SPACING. IN EACH, WE REPORT THE NUMBER OF ROWS USED (ALLOWING A ROUGHLY SQUARE CORE AREA), THE LENGTH OF THE LONGEST ROW, THE PERCENTAGE OF UNUSED CORE AREA, AND THE HALF PERIMETER WIRE LENGTH (USING THE CENTERS OF CELLS FOR PIN LOCATIONS)

| | Cell Row Spacing | | | | No Spacing | | | |
|---|---|---|---|---|---|---|---|---|
| Benchmark | Rows | Row Length | White Space | Wire Length | Rows | Row Length | White Space | Wire Length |
| Fract | 6 | 1313 | 1.3% | 65192 | 8 | 976 | .04% | 47755 |
| Struct | 21 | 4769 | 1.9% | 755176 | 29 | 3472 | 2.4% | 516236 |
| Primary1 | 17 | 5141 | .90% | 1044291 | 23 | 3610 | 3.6% | 846822 |
| Primary2 | 22 | 10501 | 1.3% | 3781467 | 39 | 6150 | 5.2% | 3007493 |
| Biomed | 44 | 10513 | 2.7% | 3403408 | 62 | 7536 | 3.7% | 2735042 |
| Avqsmall | 79 | 9497 | 3.7% | 5653141 | 112 | 6960 | 7.7% | 4347417 |
| Avqlarge | 83 | 9777 | 1.3% | 6210026 | 118 | 7648 | 2.7% | 4921154 |
| Industry1 | 24 | 3075 | 2.6% | 1625821 | 34 | 2170 | 2.6% | 950372 |
| Industry2 | 69 | 15449 | 6.5% | 15627343 | 98 | 10712 | 4.9% | 10454680 |
| Industry3 | 52 | 28153 | 3.8% | 45960568 | 74 | 20344 | 6.8% | 33832792 |
| Golem3 | 117 | 32329 | 2.2% | 89159456 | 176 | 21792 | 3.6% | 78544320 |

the placement output file. The results provided within the benchmark distribution can be replicated easily.

The scaling of these benchmarks is the same as those of TimberWolf files (from which our translations are derived). This allows integer cell sizes and locations. As the placement formats evolve, we anticipate that "benchmark units" will be added to the specification.

### D. Open Source Research and New Metrics

Industry researchers are far more interested in routability, delay optimization, power reduction, and a number of other topics; wire length is only an indirect measure of placement quality. It may initially seem somewhat surprising that academic work has focused on wire length metrics, but a brief consideration of the constraints of academic research make this situation understandable.

Much of the research in academia is performed by graduate students, who have only a few years to complete their work. Development of an effective approach toward placement with consideration of modern constraints requires expertise in electrical engineering, system architecture, circuit design, algorithms, data structures, and software engineering. This is normally well beyond the capabilities of small groups of graduate students.

In contrast, commercial tool vendors can have large teams of experienced designers and software developers. The equipment budgets and salaries far exceed those of academic groups, and the success of individuals within the team depends on the success of the entire group. It would be unreasonable to expect an academic group starting from scratch to be able to produce tools that approached the robustness and complexity of tools from a commercial group.

If academic groups are to develop sophisticated tools, and to address important industry problems, a foundation is required. The "open source" paradigm of the bookshelf helps lay this foundation: two competitive placement tools (*Feng Shui* and *Capo*) are available as both source code and executables, and research groups are encouraged to use these as part of new work. Tools for other steps in a practical design flow are under development. A number of research groups are currently using the bookshelf forum to discuss modifications and enhancements to source code, problem formulations, and file formats. In other instances (most notably the Linux kernel and GNU software tools), an open source environment has allowed the construction of software that would never be possible otherwise; our hope is that the bookshelf can facilitate similar success for design automation tools.

Placement tools available on through the bookshelf have already benefitted from the open source paradigm. Using the original source code for *Capo*, we were able to make small modifications that resulted in a 6% reduction in wire length on the Golem3 benchmark [25].

Consideration of current and emerging objectives is also simplified by the open source paradigm. With open and active discussion, new objectives can be considered, and interpretations of how to measure results can be well known. Rather than attempting to reinvent tools for old objectives, research groups will be able to focus on new problems.

## VII. CONCLUSION

Circuit placement is a central problem in VLSI design automation. While fabrication technology has advanced at a breakneck pace, design tools have had a difficult time keeping up.

In this paper, we have shown that there is wide variation in the interpretation of placement benchmarks. This has resulted in vast differences in reported results, with many instances of "apples" being inadvertently compared to "oranges." The placement problem is extremely complex, and as we have shown, many research groups have made assumptions that, while reasonable, differ substantially from the assumptions of others.

Improvement to design automation research is of profound importance. It is difficult to imagine how this will occur if we cannot accurately evaluate or compare approaches. One of the purposes of publishing research is to enable others to understand the merit of an approach, and to make subsequent comparisons. Without a common interpretation of the available benchmarks, the entire research process is undermined.

While we have focused primarily on wire length evaluation here, we note that there are far greater difficulties with performance driven design. Delay models, device parameters, determination of the longest path, and interpretations of clocking schemes, can all make significant impact on results. For example, the delay reported in [17] is *a factor of 10* lower than the result of [21]. Other areas of VLSI CAD research are likely to have similar problems: [2] focused on hypergraph partitioning, and makes a number of suggestions which could very easily apply here. As fabrication technology advances, and the size of the "design gap" increases, improvements to the reporting and evaluation of results becomes increasingly important. Our main suggestions are the following.

### A. Distribution of Placement Results

We would encourage research groups working on placement to make their placement output available (perhaps through the web). This is common practice in many other academic areas, and we expect that this would reduce ambiguity in reported results substantially. Distribution of results requires very little effort.

As part of the bookshelf [9] effort, placement results from a number of tools have been made available. In addition, both source code and executable versions of some placement tools are also available, allowing independent verification of results.

### *B. Standard Measurement*

We would also encourage the adoption of a standard for row spacing, number of rows, and method of half-perimeter wire length computation. In particular, we suggest that researchers use the formats and metrics described here. While there may be disagreement on what the "best" metrics are, there is general consensus that convergence on a single metric is far more important that the actual details of the metric. We have provided two versions of the benchmark files, allowing reporting of results using both no row spacing, and spacing equal to cell height. The GSRC Bookshelf standard cell placement slot includes a tool to determine half perimeter wire length, given a properly formatted placement result.

It should be stressed again that half-perimeter wire length should not be considered as the only measure of placement quality. Issues such as signal delay, power consumption, crosstalk, and routability are perhaps more important, but are also quite difficult to measure accurately. It is our hope that if some convergence can be obtained on half-perimeter wire length measures, we can extend this success to other metrics as well.

### *C. New Benchmarks*

Industry groups have been hesitant to release current designs, citing competitive advantage issues. We would suggest that while some competitive advantage might be lost, guidance and benchmarks from industry groups could improve the state of CAD as a whole.

Recently, a number of industrial partitioning benchmarks [1] have been adapted into standard cell placement benchmarks [22]. While these new benchmarks lack information such as signal directions, cell functionality, etc., they are relatively large, and are also free of wide ranging interpretation. We encourage research groups to report results on these benchmarks, using the row spacings, pad placements, and distance metrics provided. These benchmarks are available through the GSRC Bookshelf standard cell placement slot.

### *D. Open Source Research*

The complexity of modern design problems makes it impractical for an academic group to be competitive using only "in-house" software. Instead, we suggest that the open source paradigm should be embraced, with software contributions being made to strengthen the shared resources of the research community, and to allow the construction of a complete design flow. Open and informal discussion of software and benchmarks within the community should reduce the occurrence of interpretation differences, and improve the general quality of research.

While academic groups may lose some "competitive advantage" by making research results available in source code form, we suggest that the traditional "proprietary" approach is only successful in the short term.

## ACKNOWLEDGMENT

This paper is, in some respects, a summary of previous work. As such, it would have been impossible to complete without the cooperation and assistance of a number of people. The author would like to thank Prof. N. Abu-Ghazaleh, P. Agarwal, Dr. C. Alpert, Prof. R. Bazylevych, A. E. Caldwell, Prof. J. Cong, Prof. S. Dutt, Dr. H. Eisenmann, B. Halpin, Dr. D. J.-H. Huang, Prof. A. B. Kahng, Prof. G. Karypis, F. Khundakjie, Prof. C.-K. Koh, Prof. J. Lillis, I. L. Markov, A. Papadimitriou, Prof. M. Pedram, Dr. B. Reiss, Prof. M. Sarrafzadeh, Prof. C. Sechen, R. Smey, Dr. B. Swartz, A. Vidyarthi, Dr. D. Xu, X. Yang, and M. C. Yildiz, as well as the anonymous reviewers, for their assistance and candor.

## REFERENCES

[1] C. J. Alpert, "The ISPD98 circuit benchmark suite," in *Proc. Int. Symp. on Physical Design*, 1998, pp. 80–85.

[2] A. E. Caldwell, A. B. Kahng, A. A. Kennings, and I. L. Markov, "Hypergraph partitioning for VLSI CAD: Methodology for heurisitc development, experimentation and reporting," in *Proc. Design Automation Conf.*, 1999, pp. 349–354.

[3] A. E. Caldwell, A. B. Kahng, and I. L. Markov, "Can recursive bisection alone produce routable placements?," in *Proc. Design Automation Conf.*, 2000, pp. 477–482.

[4] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout," *Integr. VLSI J.*, vol. 21, pp. 1–94, 1996.

[5] A. E. Dunlop and B. W. Kernighan, "A procedure for placement of standard-cell VLSI circuits," *IEEE Trans. Computer-Aided Design*, vol. CAD-4, pp. 92–98, Jan. 1985.

[6] H. Eisenmann and F. M. Johannes, "Generic global placement and floorplanning," in *Proc. Design Automation Conf.*, 1998, pp. 269–274.

[7] W. C. Elmore, "The transient response of damped linear networks with particular," *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, Jan. 1948.

[8] H. Etail, S. Areibi, and A. Vannelli, "Attractor-repeller approach for global placement," in *Proc. Int. Conf. Computer-Aided Design*, 1999, pp. 20–24.

[9] GSRC. Bookshelf slot. [Online]. Available: http://www.gigascale.org/bookshelf.

[10] T. Hamada, C.-K. Cheng, and P. M. Chan, "An efficient multilevel placement technique using hierarchical partitioning," *IEEE Trans. Circuits Syst.*, vol. 39, pp. 432–439, June 1992.

[11] D. J.-H. Huang and A. B. Kahng, "Partitioning based standard cell global placement with an exact objective," in *Proc. Int. Symp. Physical Design*, 1997, pp. 18–25.

[12] S.-W. Hur and J. Lillis, "Mongrel: Hybrid techniques for standard cell placement," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 165–170.

[13] InternetCAD Inc.. Benchmark results. Results compare with TimberWolf 7.0. [Online]. Available: http://www.internetcad.com.

[14] J. Kleinhans, G. Sigl, F. Johannes, and K. Antreich, "Gordian: Vlsi placement by quadratic programming and slicing optimization," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 356–365, Mar. 1991.

[15] T. Koide, M. Ono, S. Wakabayashi, Y. Nisimaru, and N. Yoshida, "A new performance driven placement method with the elmore delay model for row based vlsis," in *Proc. Asia South Pacific Design Automation Conf.*, 1995, pp. 405–412.

[16] K. Koźmiński, "Benchmarks for layout synthesis—Evolution and current status," in *Proc. Design Automation Conf.*, 1991, pp. 265–270.

[17] S.-L. Ou and M. Pedram, "Timing-driven placement based on partitioning with dynamic cut-net control," in *Proc. Design Automation Conf.*, 2000, pp. 472–476.

[18] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.

[19] M. Sarrafzadeh and M. Wang, "NRG: Global and detailed placement," in *Proc. Int. Conf. Computer-Aided Design*, 1997, pp. 532–537.

[20] W.-J. Sun and C. Sechen, "Efficient and effective placement for very large circuits," *IEEE Trans. Computer-Aided Design*, vol. 14, pp. 349–359, Mar. 1995.

[21] W. Swartz and C. Sechen, "Timing driven placement for large standard cell circuits," in *Proc. Design Automation Conf.*, 1995, pp. 211–215.

[22] M. Wang, X. Yang, and M. Sarrafzadeh, "Dragon2000: Standard-cell placement tool for large industry circuits," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 260–263.

[23] Y.-Y. Yang and C.-M. Kyung, "HALO: An efficient global placement strategy for standard cells," *IEEE Trans. Computer-Aided Design*, p. 1024, 1992.

[24] M. C. Yildiz and P. H. Madden, "Global objectives for standard cell placement," in *Proc. GLSVLSI*, 2001, pp. 68–72.

[25] ——, "Improved cut sequences for partitioning based placement," in *Proc. Design Automation Conf.*, 2001, pp. 776–779.

[26] K. Zhong and S. Dutt, "Effective partition-driven placement with simultaneous level processing and global net views," in *Proc. Int. Conf. Computer-Aided Design*, 2000, pp. 254–259.