

# Global Routing: Metrics, Benchmarks, and Tools

Jurjen Westra\* Patrick Groeneveld\*  
jwestra@ics.ele.tue.nl patrick@ics.ele.tue.nl

\* Eindhoven University of Technology

Tan Yan†  
yant@env.kitakyu-u.ac.jp

† U. Kitakyushu

Patrick H. Madden‡  
pmadden@acm.org

‡ SUNY Binghamton

## ABSTRACT

With the increased contribution of interconnect on system delay and power consumption, it has become more critical to optimize interconnect wires. A key step in the routing of a large circuit is global routing. Surprisingly, however, there is no consensus on how global routing results are to be evaluated, nor on what constitutes a good benchmark for comparison.

In the past few years, the lack of good metrics and benchmarks in placement has been addressed. There are now widely accepted tools to evaluate placement results, and a few suites of benchmark circuits that are commonly used by placement research groups. Our objective in this paper is to obtain comparable metrics and benchmarks for global routing. We recommend specific metrics to allow meaningful comparisons, adapt widely used placement benchmarks for global routing applications, and present a set of tools to aid researchers in developing and evaluating routing approaches.

## 1. INTRODUCTION

Integrated circuits are continually growing in size, keeping pace with Moore's law. With transistor counts reaching the 1 billion mark soon, a modern chip constitutes a design challenge of unprecedented scale in human history. To handle this exponential increase in complexity, designers have relied on improved algorithms and tools. For some phases of the design process, the improvements have been easy to quantify. In circuit placement, for example, there have been measurable advances in the speed, scalability, and quality of results. For the interconnect routing, however, progress has been much more difficult to gauge.

Missing from routing research are common benchmarks and metrics. In placement, most active groups have used the MCNC[16], GSRC[11], or PEKO[8] benchmarks. Common methods for evaluating results have been established[19, 1], and many academic research groups freely exchange binary source code versions of their tools and placement results. By contrast, the only routing benchmark that could be considered "well known" is the antiquated "Deutsch's difficult example,"[9] a channel routing problem proposed nearly thirty years ago. To the best of our knowledge, the only instance of two separate groups reporting global routing results on the same set of benchmarks is [15] and [12].

Modern physical design flows employ a 'stepwise refinement' approach: the design is a refinement process of many smaller steps, rather than a few big steps. Both placement and routing are first performed at the global level using coarse models, and then refined using detailed placement and routing tools. The latter determine the exact locations and wire geometries using accurate models. This gradual transition enables the tool to respond to parasitics at many levels, and to take multiple objectives into account.

The physical synthesis tool flow consist of global placement fol-

lowed global routing, logic re-optimization, detailed placement, again global routing and finally detail routing. Global routing - the subject of this paper - performs the coarse routing of all nets in the design. It serves a number of purposes in the design flow:

1. It **reduces the complexity** of the routing task by roughly two orders of magnitude. The global router achieves this significant speed-up mainly by reducing the resolution of the routing model. After global routing the chip is partitioned into numerous small regions that are routed separately at full resolution by a detailed router.
2. It feeds realistic **estimates of the parasitic wire delay** into the logic optimization tools. The resulting path timing causes the optimization tools to address the actual critical paths.
3. It controls wire **congestion** by spreading the wiring load more evenly over the chip and between the routing layers. The subsequent detailed router should be most likely to complete all connections. This is called *routerability*.

Items 1 and 2 are well researched and understood, but controlling congestion turns out to be quite hard in practice. In this paper, we present metrics, benchmarks, and tools to compare global routing results from different algorithms. First, we analyze the global routing problem itself. Based on this analysis, we formulate metrics for the evaluation of global routing results. We then present a set of global routing benchmarks, as well as a set of tools that can be used to produce "reference results." The benchmarks are relatively large, available in a number of different formats, and are difficult enough to be challenging while still having feasible solutions. They are based on benchmarks widely used for placement.

## 2. THE GLOBAL ROUTING PROBLEM

In this subsection we will present a generic global routing problem representation. Such representation is used by more routers in some form. We will first address the modelling issues for real-life applications and the present the prior art.

### 2.1 Graph representation of the routing area

The global routing problem is formulated on a regular grid graph (see Fig. 1) where each vertex represents a small rectangular area in a layer. The area is referred to as global routing cells or *g-cells*<sup>1</sup>.

Each of the routing layers is assigned a preferred routing direction that alternates between horizontal (H) and vertical (V). Typically the bottom layer is oriented horizontally, resulting in a HVHVHV.. scheme. A vertex is generated for each layer of a g-cell, with an

<sup>1</sup>G-cells are also known as *tiles*, *bins* or *buckets*.

edge connecting the vertices in adjacent layers. This edge (in the z-direction) represents a via.

In horizontal layers, horizontal edges connect the left and right neighboring vertices, while vertical edges connect the bottom and top neighboring vertices in vertical layers. Occupying these edges will correspond to a wire segment in the layout.

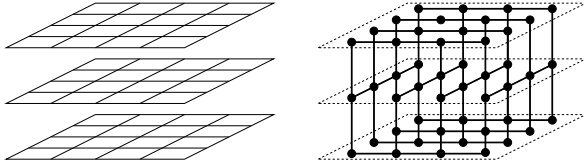


Figure 1: Grid graph model for global routing.

A g-cell represents a fixed number of available routing tracks in each layer. In the grid graph model the number of routing tracks is modelled by a capacity number on the corresponding edge called *supply*. Obstacles and pre-routed wires such as clocks reduce the supply of an edge. A global routing algorithm finds paths through this grid graph for each net that connect the g-cells of the pins. If the path uses an edge, the *demand* number on the edge is increased. The multitude of paths in a g-cell will be implemented by a detailed router in the final step.

If wiring demand exceeds supply, the detailed routing is unlikely to implement a design rule correct wire pattern. This is the most common way to detect wire ‘congestion’. If the wiring demand is less than supply, however, the g-cell is likely to be routable. It is the task of the global router to maximize ‘routability’ while optimizing other objectives. In the next section we will discuss all objectives in more detail.

## 2.2 G-cell modelling accuracy

Whether the detailed router will succeed cannot be accurately predicted based on the supply and demand numbers. The grid graph model at the global routing is too inaccurate to model the exact routing environment. The blocking effect of individual wires is not properly modelled in a grid graph that lumps multiple tracks in a single edge. Another inaccuracy is illustrated by Fig. 2 where the actual locations of the pins inside a g-cell causes the density to exceed the limits.

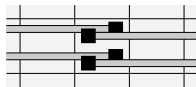


Figure 2: The interface edge between these two adjacent g-cells has a supply of two. The two wires crossing cause the demand to be equal to the supply. The local density in the center tile is 4, which makes it unlikely to be a routable g-cell.

Especially near routing blockages the capacity is often less than the supply numbers suggest. This modelling deficiency is compensated to a large extent by aggressive rip-up and re-routing in the detailed router. It is often able to produce a DRC-correct result in over-congested regions at the expense of a considerable run-time overhead. Tuning and modifying the routing track supply numbers has a significant effect on routability. Therefore this receives considerable attention in most commercial routers.

Most global routing algorithms focus on the routing capacity on the edges *between* g-cells, effectively ignoring obstacles and routing constraints within a g-cell. By reducing the size of a g-cell the modelling of obstacles in the grid graph is more accurate. In practice, g-cells are (almost) square with the height of a standard cell row.

This is equivalent to 8 to 10 routing tracks. Sizing of g-cells in this manner is common in industry applications, and also allows for the power and ground rails to be accounted for easily.

The latest IC implementation technologies have many layers (up to 9) of copper wiring. The wide power wires and macro modules block the wiring supply in a subset of these layers. It has therefore become infeasible to model the routing resource as a two-dimensional grid graph in which the edges represent all layers with a specific direction of preference. The global router must be aware of the routing layers and must be able to perform proper layer assignment.

Another modelling issue results from the fact that all active components are below the bottom routing layer. Therefore all wires begin and end at the silicon surface. The two bottom-most layers are used for local wiring to connect to the standard-cell pins, or they are fixed part of the cell itself. This limited supply and high demand often results in rather dense wiring on the lower layers. The upper layers are used for longer wires, and due to the wider pitch in these layers, they have a lower capacity.

## 2.3 Common Methods

A good survey of global routing methods is given in [13]. Most academic and industrial global routers follow a two step process. First, nets are broken into two-pin segments with minimum spanning tree (MST) or Steiner minimal tree (SMT) algorithms. Second, individual tree edges are routed using path-search algorithms such as Dijkstra’s. Usually, each edge in the routing graph has a cost associated with it; this cost increases as the demand approaches the supply figure. Through repeated iterations of rip-up and reroute, paths found for individual tree edges migrate towards to lower-density portions of the routing graph. Examples of routing tools based on this approach are [18],[12].

The routing problem can also be formulated as one of multicommodity flow –for example, [21], [20], and [4]. The most recent approach is used by some industrial groups for “difficult” routing problems– in very dense designs, the approach obtains excellent results, but at a high compute cost. For less congested designs, more traditional (and faster) rip-up and reroute methods are used.

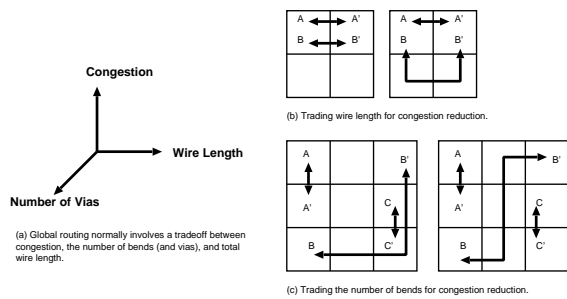
Pattern routing[15] has also been used. Rather than using a path search algorithm, routes with restricted numbers of bends were considered.

## 3. METRICS

As stated in the previous sections, the key task of a global router is to maximize *routability*. Therefore the ultimate test is to perform a full detailed routing, and measuring the number of remaining violations. Though accurate, it is also slow and very dependent on the specific detailed router that was used in the flow. Moreover, in academic environments a state-of-the-art detailed router might not be available at all. Therefore we aim for metrics that capture the behavior of a ‘typical’ detailed router. In other words we need metrics for ‘routability’.

What measurable data correlates with routability, detailed routing violations, and quality? Based on our experience, we can enumerate a few:

- **Total wire length** is a first order measure of quality for placement as well as routing. Whereas wire length is useful to drive a placer, it is much less significant for global and detailed routing. Since most global routers employ a Dijkstra-style[10] shortest path algorithm, routing all wires this way will anyway result in a short minimum wire length. Within all configurations with short total wire length, there can be huge



**Figure 3:** Global routing involves a tradeoff (a) of conflicting objectives. The “correct” mix frequently depends on the detail router used. (b) It is usually possible to reduce congestion by introducing detours—but this increases *average* wire length. (c) Inserting additional bends can also reduce congestion, but also introduces additional vias that may make detailed routing more difficult.

differences in terms of routability. Meanwhile a few percent additional wire length might not affect routability or timing noticeably.

- Some abstraction of the wiring **overflow**, that is, where routing demand locally exceeds the available routing resources. We will discuss this below.
- **Run time** complexity remains a relevant metric since the global router will be run early and often in the flow.
- **Wire quality**, that can be measured as:
  - The amount of *vias*. During global routing this is the amount of layer changes in a net.
  - The amount of *stacked vias* that connect between non-adjacent layers. They will block a wiring track in the intermediate layer(s), and with that, cause congestion.
  - The amount of *detouring*. This is the deviation from the shortest possible wire length.
  - The *evenness* of the usage of all routing resources. This is desirable not only for routing violation avoidance, but also to enable routers to improve the manufacturability by spacing out wire patterns.

For each of these global metrics, a smaller value is likely to result in a more easily routable design. In the following subsection we will address a few in more detail.

### 3.1 Congestion

Any edge where wiring demand exceeds supply indicates local congestion. The difference between supply and demand yields the *overflow* routing tracks. As indicated in the previous section, detailed routers are likely to solve a small number of overflows by “borrowing” capacity from neighboring layers or g-cells. The larger the overflow, however, the more unlikely it is that the detailed router will succeed. Also local clusters of overflowed edges will make it more unlikely that the detailed router can borrow capacity from neighboring areas.

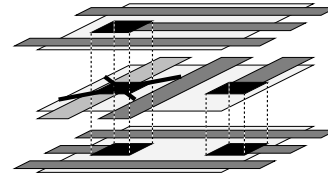
We propose to report the sum of the overflowed routing tracks over all edges as an indication of congestion. As a measure of the evenness of the track overflow the maximum of all edge overflows can be reported.

Overflow and congestion are interesting on a layer-by-layer basis. If a connection needs to be made in the horizontal direction,

for example, the detailed router may insert additional vias to gain access to a layer with the same preferred direction. Obviously, horizontal routing resource cannot be exchanged freely with vertical resource. In many industrial designs, the number of layers, track pitch, and circuit placement result in situations where there is a mismatch between routing demand and routing resource for a particular direction.

### 3.2 Bends and Vias

Vias connect adjacent copper interconnect layers. They are required to reach the different levels of interconnect, and at every bend in the route. In the global routing graph they correspond with edges in the z-direction. Compared to wire segments edges, vias have higher resistance, lower yield, and significantly larger process spread. Therefore the number of vias should be minimized. Another less obvious issue, which has a significant impact on the routability of detailed routing is illustrated in Fig. 4. Vias that connect non-adjacent layers can “blow holes” in the intermediate layers, reducing routing capacity. This effect is not modelled in the graph.



**Figure 4:** Stacked vias make holes in intermediate layers.

For uncongested designs, one can expect relatively few bends and vias. As congestion increases, routes start to meander to avoid congested areas which results in additional bends. If the number of bends is high without heavy congestion, it is an indication that the routing tool is performing incorrectly. We propose to track are the average number of vias per wire.

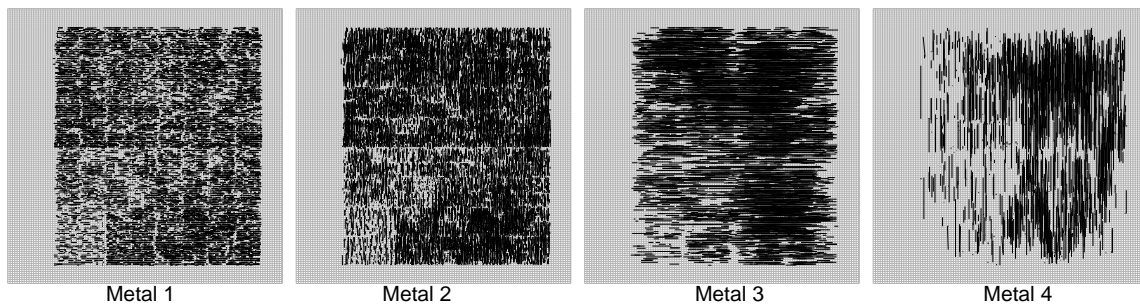
### 3.3 Measuring detours

A detour is a net path that does not have minimum wire length. Such detour can be triggered by obstacles, or by the need to spread wires to avoid congestion. The distribution of detours is relevant, since they introduce a deviation from a placement-based delay estimation. To avoid this ‘surprise factor’, the detour must be relatively short compared to the total length of a net path.

Considering the distribution of detour lengths is useful in improving the performance of a routing tool. If there are only a few detours, but they are relatively large, this is problematic for timing optimization. Further, if detours are observed in an uncongested design, this is an indication that the routing tool is not operating properly. Generally, it will not be possible to publish detour distributions for *all* benchmarks, but we encourage researchers to look at them and report the trend. Anomalies should be properly reported and explained. In order to be able to judge design difficulty and tool performance, we propose to report the total detour length as a percentage of total wire length.

While wire length is a primary objective of many placement tools, it is a less valuable metric in evaluating routing methods. Given that a placement is fixed, the detour metric mentioned above is an indication of how much routing demand is increased during global routing. As detours may be required to avoid congestion, we would argue that it is better to track how much detour has been introduced to meet a congestion objective.

The speed of a global routing method is a significant concern. Methods such as [4] are known to produce excellent results, at the expense of a considerable CPU-effort. Fast global routing allows



**Figure 5:** Included with the released benchmarks is a reference router implementation; it performs layer assignment of the Steiner tree edges, to minimize the total number of vias while equalizing routing demand across multiple layers. Each edge is then routed with a Z-shaped path, with optimization performed by a traditional rip-up and reroute method. The layer assignment results in shorter routes being placed on lower metal layers, reducing the number of vias blocking intermediate layers.

for iteration between placement, device sizing and routing, which is vital for the convergence of the design flow.

### 3.4 Adoption of New Metrics

In circuit placement, there has been an evolution and refinement of metrics over the past few years. Issues related to row spacing, pin locations, and bounding box computation methods have been debated. While not all groups are satisfied with the metrics, and there continue to be changes, the wide differences in reported results[19] have been minimized.

We expect a similar evolution in routing—not all of our proposed metrics will be adopted immediately, and we anticipate that there will be further refinement.

## 4. GLOBAL ROUTING BENCHMARKS

In this section, we describe our global routing benchmarks in detail. We have a number of specific motivations for proposing that these benchmarks be widely used.

First, the benchmarks have an *industrial background*. They are derived from a relatively recent circuits developed by IBM. Charles Alpert released a set of partitioning benchmarks[5] that were anonymized versions of the circuits; these benchmarks were then transformed into placement benchmarks[23]. The vertices in the netlists have been mapped to a widely used Artisan standard cell library. Second, the circuits are *widely used*. The benchmarks have been used by a number of research groups to perform placement experiments, and the tools Dragon[22], Capo[7], Feng Shui[3], APlace[14], and mPL[8], as well as a number of other placement tools, have all reported results on these circuits. Third, there are *established routing results* for the benchmarks[17]: in particular, the commercial tool *Cadence WarpRoute* has been used by many placement groups to evaluate their placement results.

To move from a “placement” benchmark a “global routing” benchmark, we use a specific placement for the net list. The placements, reported in [17], are obtained from *mPL-R*, with post-processing by a white-space allocation technique. *Cadence WarpRoute* has obtained 100% global and detail routing success for these placements—this shows that an acceptable solution exists. For many other placements, however, *WarpRoute* cannot route successfully—this indicates that the problems are also non-trivial, and without a routability-driven placement approach, acceptable solutions might not exist.

Most importantly, the benchmarks have not been “tuned” in any way to favor any specific approach. The benchmarks are influenced by a wide range of industrial and academic research groups, and should be relatively free of any anomalies that would skew comparisons. We suggest that when reporting results using these bench-

marks, *the complete set should be used* to avoid the chance that an abnormal behavior on a subset of the benchmarks would skew an evaluation.

The benchmarks, the size of the global routing grids, and a number of other details are summarized in Table 1. There are two variants for each net list: the “easy” version has a larger grid, giving more opportunity to reduce routing congestion.

### 4.1 Benchmark Formats

The benchmarks are available in three formats; the first allows for complete global and detail routing, and is the most desirable version from a benchmarking perspective. The complexity of the format, however, presents a barrier to academic research.

- *LEF/DEF:*

To enable complete routing of the benchmarks, the first (and preferred) format we present is in Cadence LEF/DEF. These files provide sufficient information for both global and detail routing, and we include with the benchmarks the result obtained by Cadence WarpRoute.

As LEF/DEF is a widely used format, it should be relatively easy for these benchmarks to be converted for use by other tools. We have used this format to migrate designs into commercial tool flows other than Cadence.

- *Labyrinth Format:*

Parsing LEF/DEF is non-trivial; to prevent the benchmark format from being a barrier to academic research, we have developed a version which uses the simpler format of the tool *Labyrinth*[15]. A shortcoming of this format is that it ignores layer assignments—this makes accurate measurements of the number of vias required difficult.

- *Steiner Format:* A third variant of the benchmarks includes a decomposition of multi-pin nets into two-pin edges, using a Steiner tree heuristic[6]. The trees decompositions include assignments of segments to specific routing layers, such that the area demand on each layer is equalized (compared to available resources), and the total number of vias is low. In this format, a tool must simply find a path for each two-pin connection in a three-dimensional routing graph. The format is trivial to parse.

The most versatile format is *LEF/DEF*, but many academic research groups may be unable to invest the time needed to process

Bench mark	GR Grid		Length		Vias	Usage				Max Demand			
	X	Y	MST	SMT		L1	L2	L3	L4	L1	L2	L3	L4
Easy Benchmarks													
ibm01	118	139	.697	.663	77400	20.26	33.31	28.44	17.99	11	14	11	8
ibm02	134	158	1.73	1.74	148436	19.99	25.01	30.33	24.67	13	16	17	16
ibm07	195	229	4.20	4.03	273524	18.69	33.65	28.87	18.79	14	17	17	14
ibm08	202	238	4.69	4.50	292151	20.78	30.84	33.03	15.35	16	16	17	10
ibm09	205	242	3.78	3.65	269868	20.03	33.34	28.87	17.76	14	17	17	13
ibm10	263	309	7.18	6.93	391234	19.55	33.03	31.42	16.02	17	17	16	13
ibm11	232	273	5.44	5.27	332560	19.46	33.32	30.74	16.50	17	17	17	12
ibm12	282	333	10.3	9.98	506034	17.89	35.66	28.40	18.07	17	17	17	11
Hard Benchmarks													
ibm01	116	137	6.76	6.42	67444	20.08	33.85	27.95	18.12	11	15	10	7
ibm02	131	154	1.80	1.71	153986	19.33	21.46	30.89	28.33	14	11	17	17
ibm07	190	224	4.04	3.88	238331	18.17	33.44	27.31	21.09	14	17	16	15
ibm08	197	232	4.56	4.37	290550	20.52	33.04	30.02	16.42	16	17	17	10
ibm09	200	236	3.86	3.71	269545	20.41	32.14	30.55	16.90	15	17	17	10
ibm10	256	301	6.98	6.73	391388	19.44	33.98	29.00	17.59	17	17	16	12
ibm11	226	266	5.34	5.16	335951	19.29	33.82	29.20	17.70	15	17	17	17
ibm12	275	324	9.88	9.55	425401	19.24	31.04	35.05	14.68	17	17	17	10

**Table 1:** Summary of routing benchmarks. Each is based on a placement by *mPL-R+WSA*, and uses the routing grid specified in the original *LEF/DEF* files. All lengths are scaled by  $10^8$ . We show the total spanning and Steiner tree lengths, the number of vias required in the Steiner decomposition, the percentage of total tree length assigned to each routing layer, and maximum routing demand of any graph edge on a layer-by-layer basis. Routing graph edges have a capacity of 10 on each layer, indicating significant overflow for all designs. More detail is available in the benchmarks themselves.

the benchmarks in this format. Thus, while we would strongly encourage researchers to use the native *LEF/DEF* versions, we also support the variants that are easier to parse.

The Steiner format allows an investigation of ways to route individual tree edges, in order to minimize congestion, detour, or some combination of these. A “rip-up and reroute” routing approach can be constructed to use this format, needing only a shortest path algorithm and some method of weighting routing graph edges. The *Labyrinth* format allows experiments that can combine congestion minimization with topology optimization.

## 5. TOOLS

To aid in the use of these benchmarks, we provide a number of tools to help analyze and interpret results. A summary of features and tools available is as follows.

### 5.1 File Translation

A tool has been developed which reads in *Cadence LEF/DEF* format, and produces the *Labyrinth*[15] router format, as well as easy to parse Steiner tree decompositions. The translated versions are part of the benchmark suite, and the translation tool allows conversion of other *LEF/DEF* designs into “easier to use” formats. This tool is publicly available.

### 5.2 Reference Router Implementation

By converting to *Labyrinth* format, the publicly available academic tools *Labyrinth*[15] and *Dispersion Global Router*[12] can be used.

As neither of these tools explicitly support multiple layer routing, we have also implemented a simple multi-layer router; this tool has been used to generate the reference results shown in Table 1.

The reference router operates as follows. We first decompose all nets into multi-layer Steiner trees, using a layer assignment technique similar to the one described in [2]. Each tree edge is then routed using a path with at most two bends, on the layers assigned during Steiner decomposition. Rip-up and reroute is used to adjust the paths, which remain constrained a two-bend configuration on the pre-assigned layers.

As the routes are heavily constrained and the paths are detour-free, the solutions produced have minimal wire length and very low numbers of vias—but at the cost of higher congestion than is possible with an unconstrained router. The solutions produced are intended as a point of reference; we anticipate that considerable improvements can be made easily.

Routing results produced by the tool are shown in Figure 5.

## 5.3 Visualization and Statistical Analysis

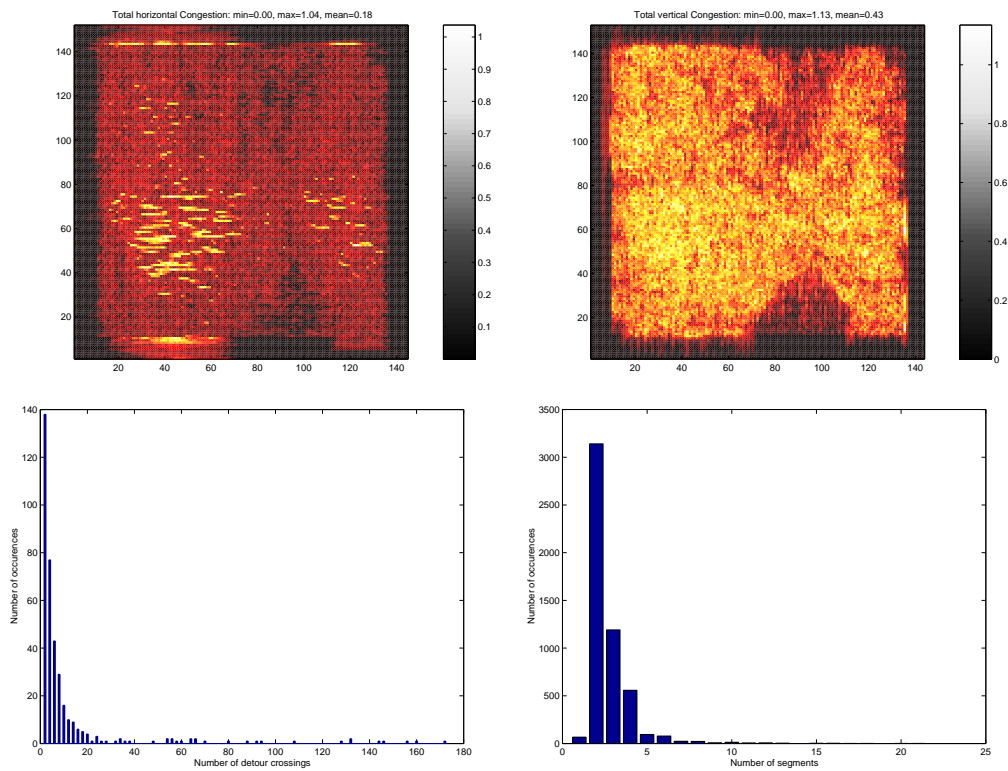
Understanding router behavior helps with the development of improved methods. To aid the research community, our tool displays routing results graphically – in both “routed wire” and “congestion map” formats. Routing results can be visualized on screen, or converted directly into PostScript format, for inclusion into technical papers.

A number of scripts and tools have been developed to analyze routing results (tracking usage per layer, detour, number of bends, and so on). At present, some of the scripts require access to either commercial CAD tools or analysis tools such as *Matlab*; we are improving the publicly available suite to encompass all the forms of analysis we discuss in the section on metrics. Samples of the statistical analysis performed by our scripts are shown in Figure 6.

## 6. CONCLUSION AND FUTURE WORK

Benchmarking is essential for research progress –without an effective method to compare two approaches, it is unclear what works, what doesn’t, and what problems need to be solved. Within industrial groups, internal benchmarks are used by development teams. In academia, however, the lack of any sort of common benchmark for routing should be alarming. We follow in the tradition of [16] and [5], proposing a specific set of benchmarks and metrics; we go further, by also providing a set of tools to perform analysis and produce reference results.

As part of our current work, we are working to improve the reference implementation to obtain better results. The tool, as well as many of the analysis methods, are integrated with a leading edge placement approach. The complete set of tools will be released shortly.



**Figure 6:** Sample analysis of the benchmark IBM01. We show congestion maps (of a commercial routing result), and histograms for detours and segment lengths.

## 7. REFERENCES

- [1] S. N. Adya, M. C. Yildiz, I. L. Markov, P. G. Villarrubia, P. N. Parakh, and P. H. Madden. Benchmarking for large-scale placement and beyond. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):472–487, 2004.
- [2] A. Agnihotri and P. H. Madden. Layer balancing for congestion reduction. In *Proc. Great Lakes Symposium on VLSI*, 2003.
- [3] A. Agnihotri, M. C. Yildiz, A. Khatkhate, A. Mathur, S. Ono, and P. H. Madden. Fractional cut: Improved recursive bisection placement. In *Proc. Int. Conf. on Computer Aided Design*, pages 307–310, 2003.
- [4] C. Albrecht. Global routing by new approximation algorithms for multicommodity flow. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 20(5):622–631, May 2001.
- [5] C. J. Alpert. The ISPD98 circuit benchmark suite. In *Proc. Int. Symp. on Physical Design*, pages 80–85, 1998.
- [6] M. Borah, R. M. Owens, and M. J. Irwin. An edge-based heuristic for Steiner routing. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 13(12):1563–1568, Dec. 1994.
- [7] A. E. Caldwell, A. B. Kahng, and I. L. Markov. Can recursive bisection alone produce routable placements? In *Proc. Design Automation Conf.*, pages 477–482, 2000.
- [8] C. C. Chang, J. Cong, and M. Xie. Optimality and scalability study of existing placement algorithms. In *Proc. Asia South Pacific Design Automation Conf.*, pages 621–627, 2003.
- [9] D. N. Deutsch. Compacted channel routing. In *Proc. Int. Conf. on Computer Aided Design*, pages 223–225, 1985.
- [10] E. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [11] GSRC. Bookshelf slot. <http://www.gigascale.org/bookshelf>.
- [12] R. T. Hadsell and P. H. Madden. Improved global routing through congestion estimation. In *Proc. Design Automation Conf.*, pages 28–31, 2003.
- [13] J. Hu and S. Sapatnekar. A survey on multi-net global routing for integrated circuits. *Integration, the VLSI Journal*, 31:1–49, 2002.
- [14] A. B. Kahng and Q. Wang. Implementation and extensibility of an analytic placer. In *Proc. Int. Symp. on Physical Design*, pages 18–25, 2004.
- [15] R. Kastner, E. Bozozzadeh, and M. Sarrafzadeh. Predictable routing. *Proc. Int. Conf. on Computer Aided Design*, pages 110–113, 2000.
- [16] K. Koźmiński. Benchmarks for layout synthesis – evolution and current status. In *Proc. Design Automation Conf.*, pages 265–270, 1991.
- [17] C. Li, C.-K. Koh, and P. H. Madden. Routability-driven placement and white space allocation. In *Proc. Int. Conf. on Computer Aided Design*, pages 394–401, 2004.
- [18] R. Linsker. An iterative-improvement penalty-function-driven wire routing system. *IBM J. Res. Dev.*, 28(5):613–624, Sept. 1984.
- [19] P. H. Madden. Reporting of standard cell placement results. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 21(2):240–247, Feb. 2002.
- [20] G. Meixner and U. Lauther. A new global router based on a flow model and linear assignment. In *Proc. Int. Conf. on Computer Aided Design*, pages 44–47, 1990.
- [21] E. Shragowitz and S. Keel. A global router based on a multicommodity flow model. *Integration, the VLSI Journal*, 5:3–16, 1987.
- [22] M. Wang, X. Yang, and M. Sarrafzadeh. Congestion minimization during placement. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 19(10):1140–1148, Oct. 2000.
- [23] M. Wang, X. Yang, and M. Sarrafzadeh. Dragon2000: Standard-cell placement tool for large industry circuits. In *Proc. Int. Conf. on Computer Aided Design*, pages 260–263, 2000.