Congestion Reduction During Placement Based on Integer Programming

Xiaojian Yang Ryan Kastner Majid Sarrafzadeh

Computer Science Department, University of California, Los Angeles, CA 90095
xjyang,kastner,majid@cs.ucla.edu

Abstract

This paper presents a novel method to reduce routing congestion during placement stage. The proposed approach is used as a post-processing step in placement. Congestion reduction is based on local improvement on the existing layout. However, the approach has a global view of the congestion over the entire design. It uses integer linear programming (ILP) to formulate the conflicts between multiple congested regions, and performs local improvement according to the solution of ILP. Experiments show that the proposed approach can effectively reduce the total overflow of global routing result. The short running time of the algorithm indicates good scalability on large designs.

1. Introduction

As VLSI system complexity continues to increase, physical design is getting more and more difficult. Traditional placement tools focus on minimizing total wirelength to obtain better routability and smaller layout area [1, 2]. Despite the pervasive use of half-perimeter wirelength objective, there is a mismatch between wirelength and congestion objectives in placement [3]; Congestion, an important objective indicating routability, has not drawn enough research attention in placement related studies. It is of value to consider routability in placement stage where the effort on congestion reduction would be more effective [4].

In [5], a routability model was proposed and incorporated in the annealing based placement. While the reduction on the congestion clearly highlights the advantage of the model, the proposed approach discards the extensive research work on wirelength minimization, and it significantly degrades placement speed. A multi-partitioning technique based on pre-determined Steiner trees was introduced in [6]. The restriction on the number of partitions confines the performance of the approach. A congestion driven placement approach was proposed in [7]. It uses area router to evaluate local congestion during placement. Several other approaches, e.g., [8, 9], also incorporate routing within placement. In practice, combining global router and placer is an effective way to improve routability, yet researchers keep studying on more efficient approaches to handle the increasing design size.

A recent study [9] shows that a post-processing technique is effective to minimize congestion because the congestion correlates with wirelength in a global view. However, reducing congestion after a wirelength-driven placement is a non-trivial problem. Traditionally, people perturb existing placement within a window around the congested area [8]. Local improvement within small windows has limited effect, whereas expanding search windows will cause interactions between congested areas, making the optimization results unpredictable.

This paper presents a novel, integer linear programming (ILP) based technique to alleviate congestion in placement. The proposed approach is used as a post-processing step during the detailed placement stage. We propose the congestion expansion technique to reduce congestion and transform the expansion problem of multiple congested areas into an ILP. To demonstrate the effectiveness of the proposed method, we use the overflow after global routing as a measurement of the placement quality.

The rest of this paper is organized as follows. Section 2 gives preliminaries. Section 3 describes the routing estimation and congestion measurement used in this work. In Section 4, we introduce an integer programming based algorithm which alleviates congestion during the detailed placement. Section 5 presents experimental results to show the effectiveness of the algorithm. We conclude in Section 6.

2. Preliminaries

A *circuit* is a hypergraph $G(C, N)$, where C is a set of cells and N is a set of nets. A *net* $e \in N$ is a subset of C which contains two or more cells. A *placement* is a set of locations for all cells within a rectangular chip area.

During the detailed placement and global routing, the core area is divided into $m \times n$ rectangular *global bins*. The boundaries of the global bins are *global bin edges*.

The *bounding box* of a net is the minimum rectangle that contains all the cells belonging to this net. The *total bounding box wirelength* of a design is the summation of the half-perimeter of the bounding box over all the nets. If we assume that the width and the height of the global bin are unit length, the *normalized total bounding box wirelength* of a design has similar definition with total bounding box wirelength, but measured by global bin grid units. The *total routed wirelength* is the sum of actual wirelength over all the nets (including each wire segment), measured by global bin grid units.

3. Congestion in Placement

3.1 Routing Estimation

To evaluate the congestion during placement, fast and accurate routing estimation is required. In this work we adopt the bounding box model from [5], as illustrated in Fig. 1.

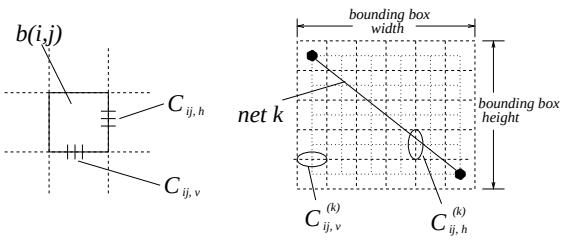


Fig. 1: Bounding box routing estimation model

For each global bin $b(i, j)$ at column i and row j, we denote

$C_{ij,h}^{(k)}$ the number of horizontal wire crossings on its right edge by net k. Similarly, we denote $C_{ij,v}^{(k)}$ the number of vertical wire crossings on its bottom edge by net k. If we use $xmin(k)$, $xmax(k)$, $ymin(k)$ and $ymax(k)$ to describe the bounding box of net k, we compute $C_{ij,h}^{(k)}$ as ($C_{ij,h}^{(k)}$ is in the symmetric form),

$$C_{ij,h}^{(k)} = \begin{cases} \frac{q(k)}{ymax(k)-ymin(k)+1} & \begin{array}{l} xmin(k) \le i < xmax(k) \\ ymin(k) \le j \le ymax(k) \end{array} \\ 0 & otherwise \end{cases}$$

where $q(k)$ is a compensation factor adopted in [5]. The existence of $q(k)$ is based on the fact that the bounding box wirelength model under-estimates the actual wiring for nets with more than three terminals. Its value depends on the number of terminals of net k. q is 1 for 2-terminal or 3-terminal nets, and slowly increases to 2.79 for nets with 50 terminals.

With the routing estimation for each net, we can calculate the total estimated number of crossings for global bin edges. For each global bin $b(i,j)$, the routing demands of its right and bottom edge are:

$$C_{ij,h} = \sum_{k=1}^{N} C_{ij,h}^{(k)} \qquad C_{ij,v} = \sum_{k=1}^{N} C_{ij,v}^{(k)}$$

3.2 Congestion Cost

We assume uniformly distributed routing tracks for the entire core area. Let Cap_h and Cap_v be the number of tracks for vertical and horizontal global bin edges, respectively. For bin $b(i,j)$, the overflow of the right edge $OF_{ij,h}$ is $max(C_{ij,h} - Cap_h, 0)$, and the overflow of the bottom edge $OF_{ij,v}$ is $max(C_{ij,v} - Cap_v, 0)$.

The congestion cost function of the design can be modeled using overflow only. A more reasonable cost function would be a combination of wirelength and overflow. In this work, we employ a combination of wirelength and quadratic function of overflow. The horizontal congestion of the design $COST_h$ is,

$$COST_h = \sum_{i=1}^{m-1} \sum_{j=1}^{n} (C_{ij,h} + OF_{ij,h}^2)$$

The vertical congestion $COST_v$ is simply the symmetric form of $COST_h$. The total congestion cost $COST$ is the sum of $COST_h$ and $COST_v$.

The total overflow OF of the layout is the sum of overflow over all the global bin edges.

To identify the congested area, we define a *congestion degree* C_{ij} for each global bin $b(i,j)$ as the average relative congestion of its four edges.

4. Congestion Reduction in Placement

4.1 Congested Region Expanding

Based on routing estimation, there are two ways to identify a congested area in placement. We can define the congested bin as the global bin with a congestion degree (described in Section 3.2) greater than a certain threshold value. Or we can define the congested bin if at least one of its edge is congested, i.e. the overflow of this edge is greater than zero. Congested region are unevenly distributed throughout the chip area.

As expected, a congested region in placement will shrink after global routing, since routers "intelligently" handle congestion. But if we change the point of view, by setting a higher threshold value in placement than in routing, the congested region is actually enlarged due to the detours in the congested spot. The tighter the routing resources, the larger the congested area.

4.2 Multiple Congested Area Expanding based on Integer Programming

The expansion of congested region suggests: (a) the congestion reduction should be performed within a larger region than the congested region; (b) certain techniques are required to handle the conflicts between the expansions of multiple congested regions. We name the congestion optimization region the *expansion area*. For a single congested region, the larger the expansion area is, the better the optimization result can be obtained because of the larger solution space. However, we should bound the expansion area since a larger expansion area requires longer running time. In addition, the expansion area of one region may overlap with that of another region if these two congested regions are close. This may cause unexpected congested regions. An arbitrative mechanism is needed to determine the expansion range for each congested region. We transform this arbitrating problem into an integer programming problem.

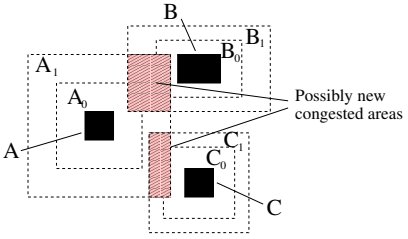
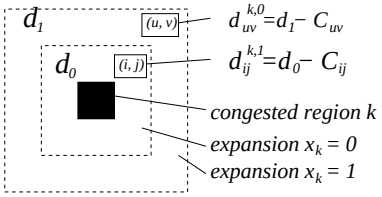


Fig. 2: Overlaps between expansion areas

Assume we have K congested regions. For each region we assign two rectangular expansion areas: a smaller one and a larger one. These expansion areas are overlapped as shown in Fig. 2. If we try to reduce congestion for one congested region without consideration of other congested regions, expansion areas may overlap and new congested region may be created.



$d_{uv}^{k,0} = d_1 - C_{uv}$

$d_{ij}^{k,1} = d_0 - C_{ij}$

Fig. 3: Two expansion areas for a congested region

The problem is to find a combination of expansion scheme for all congested regions, such that the maximum congestion over the entire area is minimized. First, we use a simplified model to describe the expansion. As shown in Fig. 3, a congested region has two expansion areas: E_0 and E_1. Let d_0 and d_1 be the expected average congestion degrees for area E_0 and E_1, respectively. Then,

$$d_0 = \frac{1}{A_0} \sum_{(i,j) \in E_0} C_{ij} \qquad d_1 = \frac{1}{A_1} \sum_{(i,j) \in E_1} C_{ij}$$

where A_0 and A_1 are the areas of E_0 and E_1, respectively.

For any global bin $b(i,j)$ in the expansion area E_0, its congestion degree before and after expansion scheme 0 are C_{ij} and d_0, respectively. For congested region k, we define the *incremental degree* for any global bin $b(i,j)$ at expansion scheme 0:

$$d_{ij}^{k,0} = \begin{cases} d_0 - C_{ij} & if\ b(i,j) \in E_0 \\ 0 & otherwise \end{cases}$$

The incremental degree for any global bin $b(i,j)$ at expansion scheme 1:

$$d_{ij}^{k,1} = \begin{cases} d_1 - C_{ij} & if\ b(i,j) \in E_1 \\ 0 & otherwise \end{cases}$$

For each congested region k, there is a corresponding binary variable x_k. x_k is 0 if the expansion E_0 is chosen, or 1 if E_1 is chosen.

The expansion scheme problem can be transformed into a binary integer linear program (ILP):

$$\begin{aligned} \text{minimize} \quad & C_{max} \\ \text{s.t.} \quad & C_{ij} + \sum_{k=1}^{K} d_{ij}^{0,k}(1-x_k) + d_{ij}^{1,k}x_k \le C_{max} \\ & i = 1,...,m \quad j = 1,...,n \\ & x_k \in \{0,1\} \quad k = 1,...,K \end{aligned}$$

where C_{max} is the maximum congestion degree over all the global bins. For each global bin there is a constraint. If a global bin $b(i,j)$ is located in the expansion area E_0 of congested region k, an item $d_{ij}^{0,k}(1-x_k) + d_{ij}^{1,k}x_k$ will be added into its constraint. If bin $b(i,j)$ is located in the expansion area E_1 but not E_0, an item $d_{ij}^{1,k}x_k$ will be generated. If bin $b(i,j)$ is located in neither of two areas, no constraint is created for the congested region k.

The transformed ILP can be optimally solved if the number of congested regions is limited. The problem solution determines the expansion scheme for each congested region. Local congestion reduction will be performed within the pre-determined expansion areas.

4.3 Congestion Reduction in Detailed Placement

Based on the congestion analysis, we propose an approach to alleviate the routing congestion in detailed placement stage. The entire flow of the approach is described by Algorithm 1.

Algorithm 1 Congestion Reduction for Multi-Congested-Area

Input: Circuit G and a detailed placement P_0,
Output: Placement P_1 with alleviated congestion

Snap cells into global bins according their current position;
for all net $n \in N$ **do**
  Do routing estimation for net n;
end for
Calculate average horizontal/vertical routing demand over all the edges: $C_{avg,h}$, $C_{avg,v}$
for all global bin $b(i,j)$, $i = 1,\ldots,m, j = 1,\ldots,n$ **do**
  Assign an estimated congestion degree C_{ij} to this bin;
end for
Identify congested regions;
Assign two expansion ranges for each congested region;
Determine expansion range for each congested region by ILP;
for all congested region **do**
  Expand this region according pre-determined expansion range;
  Do local congestion reduction within the expansion range;
end for
Create detailed placement by spreading cells within global bins.

The congestion reduction approach starts from an existing placement after wirelength minimization. The core area is divided into uniform global bins. Cells are assigned into global bins according to their current position. Based on the routing estimation model and bin congestion degree described in Section 3, we have a congestion distribution map for the current placement.

The next step is to identify the congested regions. This is accomplished by picking a congested global bin as the seed, checking the neighborhood bins and including the congested bins into the current congested region. Then we use the minimum rectangle that contains these connected congested bins as one congested region. A new seed is then picked to form the next congested region. We set a maximum area of congested regions to prevent forming too large congested regions.

Once we have all the congested regions, we assign two expansion areas for each region. The expansion scale is to the proportional of the congested region. We use $x\%$ to denote the expansion scale of a congested region, i.e., the width and height of the congested region will be $x\%$ longer than that of the original region. For each congested region, we have two expansion plans: a larger one and a smaller one. The selection between these two plans is made by formulating and solving the ILP described in Section 4.2.

The expansion area determines the range of local improvement for a congested region. The local improvement is based on cell swapping. A pair of cells are randomly chosen and swapped. Routing estimates of the nets that connect to these two cells are re-evaluated. The swap will be accepted if the total congestion cost in the expansion area is lower after swapping, and will be rejected otherwise. In order to speed up the performance, cell swapping is performed based on global bin structure, i.e. cells are located at the center of global bins. This may cause the cost function mismatch between the current placement and the final placement after resolving overlap. To limit the mismatch, row balance and bin balance are maintained during the congestion reduction.[1]

Local improvement is performed for each congested region. After a given iteration of improvement, the algorithm terminates by spreading cells within each global bin. A final detailed placement is then created.

5. Experimental Results

The proposed approach has been implemented in C. All the experiments were done in Linux environment on a PC with a 733MHz CPU. The experimental circuits are chosen from IBM-PLACE benchmarks [12]. For each circuit, we place it using a wirelength-driven placer, Dragon [11], then route it using a global router based on maze routing and rip-up and re-route [13]. Also, we perform proposed congestion reduction approach on the same placement, followed by global routing using same parameters. We compare the global routing results (overflow and routed wirelength) for placements with or without congestion reduction.

Table 1 shows the effect of congestion reduction as a post-processing. As can be seen, the congestion reduction approach considerably reduces the total overflow after global routing. For the best case among all the results, circuit *ibm05*, the total overflow turns to zero by congestion reduction. As of total routed wirelength, almost all the circuits have a shorter wirelength after congestion reduction. This indicates that wirelength is not sacrificed due to the reduction of overflow. We also noted that the total bounding box wirelength (both actual or normalized grid wirelength) with or without congestion reduction do not differ a lot. This reveals: (a) the bounding box wirelength is no longer a good metric of routability in detailed placement, and (b) our proposed approach does small perturbation on the existing placement. Finally, the short amount of running time shows that the method can scale well for large circuits.

Table 2 shows the effect of ILP based expansion technique in congestion reduction. For a given circuit *ibm*02 which contains 9

[1]We use row balance factor 0.01 and bin balance factor 0.50.

				V/H	overflow		BB-WL		grid-BB-WL		grid-routed-WL		runtime
ckt	*#cells*	grids	c/b	Cap	NCR	CR	NCR	CR	NCR	CR	NCR	CR	(seconds)
ibm01	12,036	64× 64	2.9	12 / 14	398	**309**	4.79	0	52742	0	76517	-0.9%	11
ibm02	19,062	80× 64	3.7	22 / 34	492	**292**	13.82	-0.7%	142240	-0.1%	204734	-2.4%	33
ibm03	21,924	80× 64	4.3	20 / 30	209	**181**	13.02	0	129654	0	185194	-1.8%	25
ibm04	26,346	96× 64	4.3	20 / 23	882	**778**	16.31	+0.1%	147943	0	196920	-0.7%	39
ibm05	28,146	128× 64	3.4	42 / 63	251	**0**	37.14	0	472110	0	689671	-9.8%	134
ibm06	32,185	128× 64	3.9	19 / 34	834	**540**	20.67	-0.1%	241014	-0.1%	346137	-0.9%	169
ibm07	45,135	192× 64	3.6	21 / 36	697	**686**	30.76	+0.1%	327253	-0.1%	449213	+0.4%	164
ibm08	50,977	192× 64	4.1	21 / 32	665	**654**	33.56	-0.1%	339442	-0.1%	469666	-0.2%	181
ibm09	57,746	256× 64	3.1	14 / 28	505	**268**	30.38	-0.1%	367378	-0.1%	481176	-1.2%	232
ibm10	67,692	256× 64	4.1	27 / 40	588	**383**	59.81	+1.1%	513215	+1.2%	679606	-0.3%	389

Table 1: Global routing results comparison between placement without congestion reduction (NCR) and with congestion reduction (CR). The total overflow, bounding box wirelength, normalized bounding box wirelength and normalized routed wirelength are compared. Lower overflows are shown in boldface. Wirelength with CR is relative to that without CR. Runtime is in CPU seconds. Circuit statistics are also shown, including number of cells, number of global bins, average number of cells per bin (c/b) and vertical/horizontal capacity (V/H Cap) in global routing.

single expansion				double expansion	
exp %	OF	exp %	OF	exp %	OF
10	447	60	433	10/60	412
20	457	70	439	20/70	396
30	473	80	355	30/80	436
40	427	90	457	40/90	388
50	430	100	424	50/100	380

Table 2: Comparison between single expansion and ILP based expansion for circuit ibm02. Overflow is reported for each expansion area x% in single expansion case, and for each combination of two expansion areas x_1%/x_2% in double expansion case.

congested regions, we use single expansion technique with different expansion areas from 10% to 100%. The double expansion technique is also applied with different combination of expansion areas. We use commercial ILP solver CPLEX 7.0 to solve the formulated ILP. All of these methods are tested by reporting the total overflow after global routing. As can be seen, in general the double expansion scheme produces better placement comparing with the single expansion, e.g., 20%/70% combination is better than either 20% or 70% expansion.

Experimental results and place/route tools used in this work could be found at: http://www.cs.ucla.edu/~xjyang/iccad/.

6. Conclusion

We have developed an algorithm to alleviate congestion during placement. In the congestion reduction process, routing estimation model is used to evaluate edge congestion and bin congestion. Congested spots on the design are relieved using the local improvement within a search window. Such a window size is determined by formulating and solving a ILP when dealing with multiple congested areas. Experimental results on overflow of routed design show that the proposed method is an effective way to improve design routability.

7. Acknowledgments

The authors would like to thank the reviewers for their helpful comments.

8. References

[1] W. J. Sun and C. Sechen. "Efficient and Effective Placement for Very Large Circuits". *IEEE Transactions on Computer Aided Design*, 14(3):349–359, March 1995.

[2] J. M. Kleinhans, G. Sigl, F. M. Johannes, and K. J. Antreich. "GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization". *IEEE Trans. on Computer Aided Design*, 10(3):365–365, 1991.

[3] A. E. Caldwell, A. B. Kahng, and I. L. Markov. "Can Recursive Bisection Alone Produce Routable Placements?". In *Design Automation Conference*, pages 153–158. IEEE/ACM, June 2000.

[4] A. B. Kahng, S. Mantik, and D. Stroobandt. "Requirements for Models of Achievable Routing". In *International Symposium on Physical Design*, pages 4–11. ACM, April 2000.

[5] C. E. Cheng. "RISA: Accurate and Efficient Placement Routability Modeling". In *International Conference on Computer-Aided Design*, pages 690–695, 1994.

[6] S. Mayrhofer and U. Lauther. "Congestion-Driven Placement Using a New Multi-partitioning Heuristic". In *International Conference on Computer-Aided Design*, pages 332–335. IEEE, 1990.

[7] P. N. Parakh, R. B. Brown, and K. A. Sakallah. "Congestion Driven Quadratic Placement". In *Design Automation Conference*, pages 275–278. IEEE/ACM, June 1998.

[8] R. S. Tsay and S. C. Chang. "Early Wirability Checking and 2-D Congestion-Driven Circuit Placement". In *International Conference on ASIC*. IEEE, 1992.

[9] M. Wang, X. Yang, and M. Sarrafzadeh. "Congestion Minimization During Placement". *IEEE Transactions on Computer Aided Design*, 19(10):1140–1148, 2000.

[10] J. Lou, S. Krishnamoorthy, and H. S. Sheng. "Estimating Routing Congestion using Probabilistic Analysis". In *International Symposium on Physical Design*, pages 112–117. ACM, April 2001.

[11] M. Wang, X. Yang, and M. Sarrafzadeh. "Dragon2000: Fast Standard-cell Placement for Large Circuits". In *International Conference on Computer-Aided Design*, pages 260–263. IEEE, 2000.

[12] http://www.ece.nwu.edu/nucad/ibm-place.html

[13] http://www.cs.ucla.edu/~kastner/labyrinth/