

# Congestion Minimization During Placement

Maogang Wang , Xiaojian Yang , Majid Sarrafzadeh

*Abstract*— Typical placement objectives involve reducing net-cut cost or minimizing wirelength. Congestion minimization is the least understood, however, it models routability most accurately. In this paper, we study the congestion minimization problem during placement. First, we show that a global placement with minimum wirelength has minimum total congestion.

We show that minimizing wirelength may (and in general, will) create locally congested regions. We test seven different congestion minimization objectives. We also propose a post processing stage to minimize congestion. Our main contribution and results can be summarized as below:

1. Among a variety of cost functions and methods for congestion minimization (including several currently used in industry), wirelength alone followed by a post processing congestion minimization works the best and is one of the fastest.
2. Cost functions such as a hybrid length plus congestion (commonly believed to be very effective) do not always work very well.
3. Net-centric post-processing techniques are among the best congestion alleviation approaches.
4. Congestion at the global placement level, correlates well with congestion of detailed placement.

*Keywords*— Placement, Congestion, Optimization, Minimization

## I. INTRODUCTION

**A**UTOMATED cell placement for VLSI circuits has always been a key factor for achieving designs with optimized area usage, wiring congestion and timing behavior. As technology advances, the congestion problem becomes more and more important. With the advent of over-the-cell routing, the goal of every place and route methodology has been to utilize area to prevent spilling of routes into channels. It is this overflow of routes that accounts for an increase in area. The multiple routing layers have enough routing resources to route most wires as long as there are not too many wires congested in the same region. Excessive congestion will result in a local shortage of the routing resource. In this paper, we concentrate on placement problems with fixed boundaries and little white space so that routing needs to be done in upper routing layers.

Typical placement objectives involve reducing net-cut costs or minimizing wirelength. Because of its constructive nature, min-cut based strategies minimize the number of net crossings but fail to uniformly distribute them [9]. Congestion-driven placement based on multi-partitioning

was proposed in [7]. It uses the actual congestion cost calculated from pre-computed Steiner trees to minimize the congestion of the chip, however, the number of partitions is limited due to the excessive computational load. The use of minimal wirelength as a metric to guide placement has been successful in achieving good placement. However, it only indirectly models congestion and the behavior of the router. Reducing the global wirelength helps reduce the wiring demand globally, but does not prevent existing local congested spots. It is entirely feasible for a minimum wirelength solution to require more routing resources through a region than are available. Therefore, traditional placement schemes which are based mainly on wirelength minimization, e.g., see [10], [4], [12], [1], [15], [5], [2], [14], [13] cannot adequately account for congestion.

The congestion problem in placement is not well studied. There are not many results on this problem [7], [8], [16], [11]. In this paper, we will study the congestion problem during placement. We first point out that minimizing wirelength is indeed equal to minimizing the average routing demand. Then by giving an example we show that the congestion cost could be locally inconsistent with the wirelength cost. We also establish a relationship between minimizing wirelength and minimizing congestion. Then we focus on finding a good objective to effectively reduce the congestion in the final placement. Using the congestion cost directly as the objective is not effective. The congestion cost is a badly behaved objective function because it is not sensitive to placement moves.

We tested seven congestion related objectives, experiments show that the traditional wirelength objective works the best on all testing circuits. Based on the properties of congestion minimization, we propose a two step approach to effectively produce a congestion minimized placement. The first step is a traditional wirelength minimization stage which can also reduce the congestion globally. After that, a post processing stage is used to reduce local congested spots. This two-stage minimization flow is found to be much more effective than minimizing congestion in one step or to simultaneously minimize wirelength and congestion. In the post processing stage, we experimentally tested three algorithms: a greedy cell-centric approach, a flow-based cell-centric approach and a net-centric approach. We get best congestion results by using the net-centric approach in the post processing stage. The placement produced by this new objective has on the average 36.9% less congestion than the best congestion results obtained by commonly used objectives.

The rest of the paper is organized as follows: In section II, we formally define the congestion cost. In section III, we discuss the relations between wirelength and congestion. In section IV, we show that what is a good routing estimation

This work was supported in part by NSF grant MIP-9527389.

M. Wang is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA. E-mail: mgwang@ece.nwu.edu .

X. Yang is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA. E-mail: xjyang@ece.nwu.edu .

M. Sarrafzadeh is with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA. E-mail: majid@ece.nwu.edu .

model to use in placement. In Section V, we introduce several objectives to use in the congestion minimization and compare seven different objectives. The post processing stage and algorithms to use in this stage are introduced in Section VI and the conclusion is in Section VIII.

## II. DEFINITION OF THE CONGESTION COST

Intuitively speaking, congestion in a layout means too many nets are routed in local regions. In this paper we assume that we are given a netlist that consists of a set of cells connected by a collection of nets. Each net consists of a set of pins. Cells are to be assigned a geometric location on the layout surface in the placement process. We mainly concentrate on placement problems when boundary of the chip is given and there is very little white space ( area not occupied by cells). Thus, most nets need to be routed on the upper layers. Most present-day designs follow this paradigm.

The congestion cost is defined based on the global bin concept. We partition a given chip into several rectangular regions, each of these regions is called a global bin. The boundaries of global bins are called global bin edges. Assume we have  $r$  rows and  $c$  columns of global bins. We label the global bin at  $i_{th}$  row and  $j_{th}$  column as  $B_{ij}$ . From the top left global bin, the labels are  $B_{11}, B_{12}, B_{13}, \dots, B_{ij}, \dots, B_{rc}$ . Figure 1 shows an example. In Figure 1, we have  $4 \times 4 = 16$  global bins.

The congestion is “related” to the number of crossings between routed nets and global bin edges. Each global bin has two horizontal and two vertical edges surrounding it. We will refer a horizontal global edge as  $e_h$  and a vertical global edge as  $e_v$ .

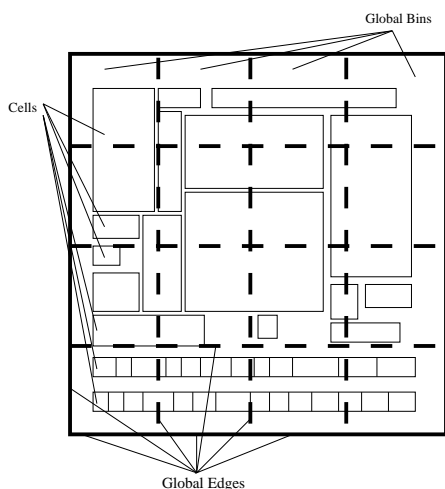


Fig. 1. Layout of a circuit and global bins.

Given a placement, all the cells and pins have fixed positions on the chip. In order to get the congestion information, we need to estimate the final routing chip. We can use a “router” to route all the nets. This router is not necessarily a detailed router. It can be a very simple global router or even a bounding box router. Obviously, the more accurate the router, the more accurate is the estimation

at the placement stage. For each global edge, there are routed nets going across it. Therefore, for each global edge  $e$ , the routing demand of  $e$ ,  $d_e$ , is defined as the number of the nets crossing  $e$ . The routing supply of a global edge  $e$ ,  $s_e$ , is a fixed value which is a function of the length of the edge and technology parameters. A global edge  $e$  is congested if and only if the routing demand (number of the crossing nets) exceeds the routing supply of that edge ( $d_e > s_e$ ). If a global edge  $e$  is congested, the overflow of  $e$  is defined as the exceeding amount of the routing demand over the routing supply of  $e$ . The overflow of  $e$  is zero if  $e$  is not congested. Congestion map produced by CAD vendors provides information on the overflow as defined in this paper. The overflow is formally described as:

$$overflow_e = \begin{cases} d_e - s_e & \text{if } d_e > s_e \\ 0 & \text{if } d_e \leq s_e \end{cases}$$

Using the above global bin and global edge notation, the *total overflow* of a placement is defined as the summation of the overflow for all global edges. The amount of total overflow reflects the amount of total shortage of routing resource in the placement. Thus a placement with less total overflow is less congested. Our experience with industry routers show that the total overflow is a good measure of congestion.

## III. CORRELATIONS BETWEEN WIRELENGTH AND CONGESTION

In order to normalize the wirelength of the nets, we use the dimension of the global bin grid as the unit length. The width of a global bin is the unit length in the  $x$  direction and the height of a global bin is the unit length in the  $y$  direction. Given locations of all pins, there are a number of ways to route all the nets. For example, we can use the bounding box, the minimum spanning tree (MST) or the Steiner tree model to estimate the actual routing. A bounding box and a MST routing model are illustrated in Figure 2.

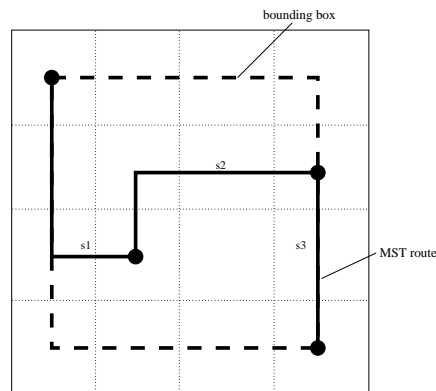


Fig. 2. A to-be-routed 4-pin net.

The congestion is not independent of the wirelength cost. Intuitively, a layout with optimized wirelength will have

less nets going through the same region, thus the congestion cost of the layout is also expected to be minimized.

**Observation 1:** Assuming cells are placed at centers of global bins, the total wirelength of a global placement is equal to the total routing demand on all global edges, i.e.,  $\sum_{\alpha} l_{\alpha} = \sum_e d_e$ , where  $l_{\alpha}$  is the estimated length for net  $\alpha$  and  $d_e$  is the routing demand for global bin edge  $e$ .

Since we are using the dimension of the global bin as the unit length to measure the wirelength, each unit length wire will cross a global edge. Thus, each unit of the wirelength will contribute to one crossing between the wire and a global edge which is by definition one unit of the routing demand. Therefore the total units of wirelength will be equal to the total units of the routing demand.

This observation shows the underlying correlations between the wirelength cost and the congestion cost. When we minimize the wirelength cost, the total amount of routing demand is minimized. Thus the average routing demand on a global edge is minimized. Given a fixed amount of routing supply which is dependent on technology parameters, the less the routing demand is, the bigger chance we will get a low-congestion layout. Based on this observation, we conclude that minimizing congestion is globally consistent with minimizing wirelength. However, these two tasks may not be consistent in local regions.

Figure 3 shows an example that minimizing congestion is not equivalent to minimizing wirelength. The sample circuit contains eight cells and four nets. Among these eight cells, four have no nets attached to them and the other four are circularly connected by four nets. Assume that the wiring supply on each global edge is one, the left part of Figure 3 shows a congestion optimal placement. In this placement, four nets are evenly distributed on the chip which result in a zero overflow (routable) solution. In wirelength optimization, we tend to put as many nets as possible into the same region. The right part of Figure 3 shows the wirelength optimized placement. Since each global bin can only contain two cells, we put four cells along with four nets into two global bins. This results in a wiring demand of two on one global edge. Since the wiring supply is only one, we have overflow of one in this placement.

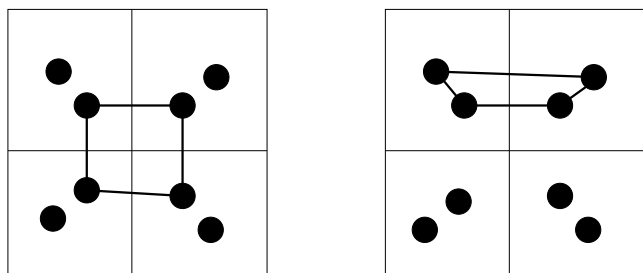


Fig. 3. Minimizing congestion is not equivalent to minimizing wirelength.

A similar trend has been observed in placement of large circuits. Figure 4 shows the two dimensional congestion map of a wirelength optimal placement for MCNC benchmark circuit Primary2. The congestion on the chip is not

balanced. Therefore there are a number of highly congested spots. When minimizing wirelength, we tend to put cells within a highly connected cluster close to each other. On the other hand, when minimizing congestion, we tend to balance all the wires to avoid local congested spots. Thus we might spread out the highly connected clusters slightly to reduce congestion. Therefore, minimizing wirelength and minimizing congestion may conflict each other in local regions. In order to get a congestion optimal placement, we might have to sacrifice wirelength.

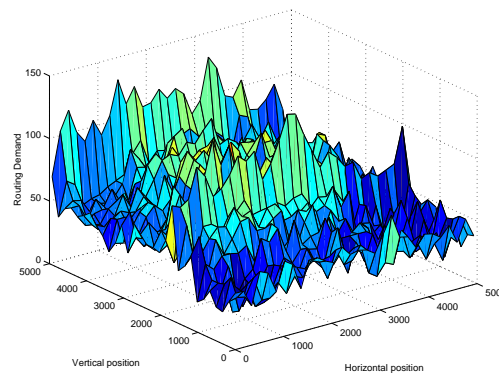


Fig. 4. Actual congestion distribution on a two-dimensional layout for Primary2.

#### IV. DIFFERENT ROUTING ESTIMATION MODELS

When we are performing minimization, we need to estimate congestion of placement incrementally. In this section, we will discuss two incremental routing estimation models, one simple model and a more accurate one (both models have been studied extensively in the past).

The first routing model can be best described as a “bounding-box model”. This model is different with the router used after placement. However, it is very simple and fast. Figure 5 shows a net which contains five terminals (represented by black solid dots in Figure 5). This method is shown in Figure 5a. Given locations of all the terminals of a net, first we find the bounding box of the net. Then the actual route will be either the upper L-shape half or the lower L-shape half of the boundary of the bounding box determined in a probabilistic manner. This method will ignore terminals in the middle of the bounding box for nets which have more than two terminals.

The second model is a real global routing model. This is the same router used after placement. This model will provide a very accurate congestion estimation during the placement stage. However, it is slower than the bounding box router. Routing is a relatively well studied problem. The Steiner tree based maze routing technique is usually used in the routing stage. We will use this router for the incremental congestion estimation.

We conduct an experiment to test if these two routing models correlate to each other. First we generate a number of different placements for the same circuit. Then we evaluate the overflows of these generated placements using



7b shows the differential curve for the overflow cost which shows the change in the real cost function. The wirelength cost can also be expressed as the summation of the number of crossing nets on all global bin edges according to Observation 1. Figure 6a shows the wirelength cost curve on each single global bin edge. Figure 7a shows the differential curve for the wirelength cost. Comparing these two differential curves (Figure 7a and 7b), the wirelength cost is much more smooth than the overflow cost because the overflow cost has a sudden jump when the number of crossing nets is around the routing supply at that global bin edge  $S$ . The real cost and the differential cost curve of the hybrid cost,  $(1 - \alpha)WL + \alpha Overflow$ , are shown in Figure 6c and 7c, respectively.

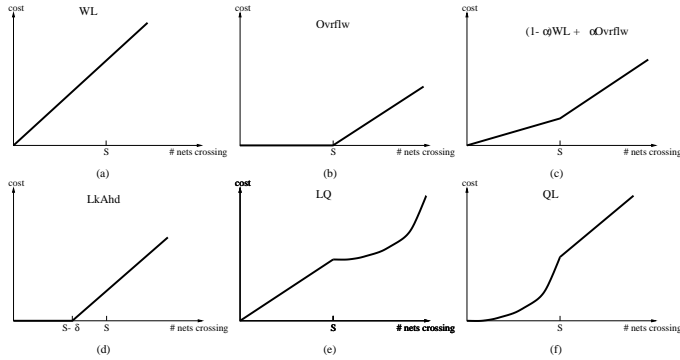


Fig. 6. Cost function vs. number of crossing nets on each global bin.

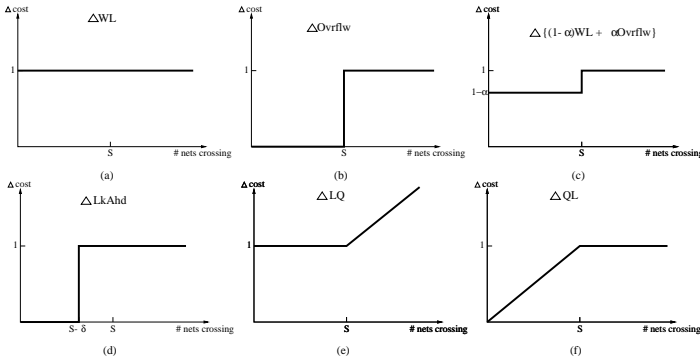


Fig. 7. Differential cost function vs. number of crossing nets on each global bin.

We know that wirelength is indirectly correlated to congestion, so it would not give us the best result for congestion. The overflow objective is a direct measure of congestion. If we use an optimal optimization technique, we should be able to get a layout with the minimum congestion. However, since the placement problem is NP-hard, no existing heuristic is perfect. Any optimization technique we use is actually a local optimization technique given finite amount of time. Thus the optimization result highly depends on the properties of the objective function. A smooth objective function will be easier for an optimization heuristic to find the global minimum.

As shown in Figure 7a and 7b, the overflow objective is not as smooth as the wirelength objective. When we

move a cell, the routing demand is changed on some global edges. However, if the routing demand before and after the change are both less than or equal to the routing supply of that edge, the overflow will not change. Therefore, the direct overflow cost may not be a very effective objective for iterative optimization techniques. By combining the wirelength and the overflow cost, the hybrid objective might be a reasonable objective to use.

Besides the three objectives mentioned above (pure wirelength, pure congestion and the hybrid objective), we also construct a couple of other objectives which we think might be good to use to reduce congestion. The differential curve of the first cost function is shown in Figure 7f. Instead of taking a sudden jump when the number of crossing nets hits  $S$ , the change of the new cost function gradually increases from 0 to 1 when the number of crossing nets changes from 0 to  $S$ . The corresponding real cost function is shown as in Figure 7f. The actual cost curve consists of two parts. The first part is a quadratic curve and the second part is a linear curve. Thus we call it a QL cost function. Similarly, we can construct another new cost, LQ cost. The differential and the real cost curve are shown in Figure 7e and 6e.

For any global edge  $e$ , the routing supply is  $s_e$ . Suppose the routing demand of  $e$  is  $d_e$  before a move and  $d'_e$  after the move. The direct overflow cost of this move will be  $\max(d_e, s_e) - \max(d'_e, s_e)$ . As we can see, if  $d_e < s_e$  and  $d'_e < s_e$ , the cost of the move will be zero. However, if  $d_e$  or  $d'_e$  is close to  $s_e$ , i.e.,  $s_e - \delta \leq d_e, d'_e \leq s_e$  where  $\delta$  is a small number, the change on  $d_e$  is still useful to evaluate the move. For example, an increase in  $d_e$  will result in a higher probability of changing the edge  $e$  from uncongested to congested in later moves; and a decrease in  $d_e$  will help the edge  $e$  stay uncongested in the future. On the other hand, if  $d_e$  and  $d'_e$  are both far less than  $s_e$ , i.e.,  $d_e, d'_e \leq s_e - \delta$ , we do not care about the change in  $d_e$  because the edge  $e$  will more likely remain uncongested in the near future. Based on this discussion, we propose another cost function called, *overflow with look-ahead*. The cost of each move is  $\max(d_e, s_e - \delta) - \max(d'_e, s_e - \delta)$  where  $\delta$  is an adjustable parameter. The differential and the real cost curve of this look-ahead cost is shown in Figure 7d and 6d. Finally, in the hybrid cost function mentioned above ( $(1 - \alpha)WL + \alpha Overflow$ ),  $\alpha$  is a constant throughout the optimization procedure. We can let  $\alpha$  be  $\alpha_T$  which changes as the optimization proceeds. Since minimizing wirelength is globally equal to minimizing congestion, we can initially let  $\alpha_T$  be zero so that the hybrid cost function is equal to a pure wirelength cost function. Then as the optimization proceeds, we gradually increase the value of  $\alpha$  so that the cost function changes gradually from wirelength to overflow. We call this cost function a time changing cost function

To summarize, we have the following seven objectives to use to reduce the congestion in a placement:

- *WL*: Standard total wirelength objective.
- *OF*: Total overflow in a placement. This is a direct measure of the congestion.

- *Hybrid*:  $(1 - \alpha)WL + \alpha OF$ , where  $0 \leq \alpha \leq 1$ .
- *QL*: A quadratic plus linear objective as described above.
- *LQ*: A linear plus quadratic objective as described above.
- *LkAhd*: Modified overflow cost with look-ahead as described above.
- $(1 - \alpha_T)WL + \alpha_T OF$ : A time changing hybrid objective which lets the cost function gradually change from wirelength to overflow as optimization proceeds.

In order to test these seven objectives, we ran eight MCNC standard-cell benchmark circuits. The characteristics of these circuits are shown in Table V. The size of the global bin grid is chosen so that each bin has roughly 5 – 50 cells.

TestCase	# Cells	# Nets	Global Bins
highway2	62	87	4×5
fract	125	163	6×6
Primary1	833	1266	8×8
Primary2	3014	3817	16×20
struct	1888	1920	16×10
biomed	6417	7052	40×50
avqs	21584	30038	20×20
avql	25114	33298	20×20

TABLE V  
TESTING CIRCUITS INFORMATION.

We can test the proposed objective with any placement heuristic. We have selected Simulated Annealing (SA). It is theoretically proved that given infinite amount of time, SA can get the global optimal result for any objective function. SA is widely used in VLSI CAD tools. The Timber-Wolf placement package [12] and the NRG placement tool [10] use simulated annealing and produce very good results on wirelength. Besides SA, other optimization techniques could be chosen as well. Results in this paper are obtained using NRG's global placer. However, the objective of this paper is to show how to improve congestion of *ANY* placement result.

For the hybrid cost function, we let  $\alpha$  be 0.2, 0.4, 0.5, 0.6 and 0.8, respectively. For the time changing cost function, we start  $\alpha_T$  from 0. Then we increase  $\alpha_T$  by 0.1 every 10 iterations of simulated annealing. Since we have about 120 iterations in total for the whole simulated annealing procedure, the value of  $\alpha_T$  will change from 0 to 1 while annealing proceeds.

Table VI shows the results for circuit *biomed*. Each row of Table VI is corresponding to one of the testing cost objectives. We run simulated annealing with each of the testing objectives. After the annealing is done, we report the wirelength and the overflow of the final placement. Table VII – XIII show the results of the rest of the testing circuits.

From Table VI – XIII, the wirelength objective is clearly the winner. The overflows produced by the wirelength are far less than the overflows produced by other

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	27885	3011	643
<b>OF</b>	57992	20400	116050
$0.8WL + 0.2OF$	53289	20982	51001
$0.6WL + 0.4OF$	56993	23399	53398
$0.5WL + 0.5OF$	58016	23768	50074
$0.4WL + 0.6OF$	59434	24954	49283
$0.2WL + 0.8OF$	62450	27063	49884
$(1 - \alpha_T)WL + \alpha_T OF$	65233	29486	47300
<b>LkAhd</b>	70346	32367	43523
<b>QL</b>	65532	27738	47426
<b>LQ</b>	67786	30846	48212

TABLE VI  
COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *biomed*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	120	12	3.8
<b>OF</b>	179	10	22.9
$0.8WL + 0.2OF$	170	26	62
$0.6WL + 0.4OF$	145	13	37
$0.5WL + 0.5OF$	159	20	59
$0.4WL + 0.6OF$	165	27	53
$0.2WL + 0.8OF$	189	41	58
$(1 - \alpha_T)WL + \alpha_T OF$	204	37	62
<b>LkAhd</b>	137	12	90
<b>QL</b>	139	2	92
<b>LQ</b>	136	9	82

TABLE VII  
COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *highway2*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	290	16	8.5
<b>OF</b>	406	23	72
$0.8WL + 0.2OF$	348	32	83
$0.6WL + 0.4OF$	511	163	182
$0.5WL + 0.5OF$	483	104	198
$0.4WL + 0.6OF$	426	80	183
$0.2WL + 0.8OF$	538	169	228
$(1 - \alpha_T)WL + \alpha_T OF$	674	272	230
<b>LkAhd</b>	339	9	351
<b>QL</b>	347	5	384
<b>LQ</b>	375	35	342

TABLE VIII  
COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *fract*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	6067	34	30
<b>OF</b>	12808	480	468
$0.8WL + 0.2OF$	8477	95	685
$0.6WL + 0.4OF$	11090	479	939
$0.5WL + 0.5OF$	12695	595	894
$0.4WL + 0.6OF$	13859	639	904
$0.2WL + 0.8OF$	14726	990	956
$(1 - \alpha_T)WL + \alpha_TOF$	15437	1062	1087
<b>LkAhd</b>	10344	249	432
<b>QL</b>	10056	179	506
<b>LQ</b>	10523	362	415

TABLE IX

COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *Primary1*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	26918	151	269
<b>OF</b>	80425	6391	9116
$0.8WL + 0.2OF$	79918	9406	17103
$0.6WL + 0.4OF$	81704	9149	17108
$0.5WL + 0.5OF$	84586	9660	17145
$0.4WL + 0.6OF$	89734	10883	17167
$0.2WL + 0.8OF$	96108	12052	17517
$(1 - \alpha_T)WL + \alpha_TOF$	100869	13055	17761
<b>LkAhd</b>	77823	5613	9267
<b>QL</b>	66086	4231	11600
<b>LQ</b>	75090	6298	10284

TABLE X

COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *Primary2*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	3397	88	234
<b>OF</b>	11047	3196	1490
$0.8WL + 0.2OF$	10850	4258	3176
$0.6WL + 0.4OF$	13565	5507	3298
$0.5WL + 0.5OF$	14958	5603	3285
$0.4WL + 0.6OF$	15104	5820	3234
$0.2WL + 0.8OF$	15705	5897	3318
$(1 - \alpha_T)WL + \alpha_TOF$	16154	5974	3240
<b>LkAhd</b>	6779	998	4248
<b>QL</b>	5839	349	4844
<b>LQ</b>	6935	989	4234

TABLE XI

COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *struct*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	9110	159	5839
<b>OF</b>	718451	130410	93381
$0.8WL + 0.2OF$	651406	117992	89283
$0.6WL + 0.4OF$	655704	118569	93330
$0.5WL + 0.5OF$	658943	118994	89081
$0.4WL + 0.6OF$	660134	119084	90385
$0.2WL + 0.8OF$	661199	119243	90469
$(1 - \alpha_T)WL + \alpha_TOF$	698035	126173	60884
<b>LkAhd</b>	711535	128970	61417
<b>QL</b>	669985	120612	59896
<b>LQ</b>	718701	130538	61840

TABLE XII

COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *avqs*.

Cost Function	wire-length	over-flow	run-time(s)
<b>WL</b>	107261	802	7934
<b>OF</b>	879751	160520	113085
$0.8WL + 0.2OF$	832858	153260	110778
$0.6WL + 0.4OF$	838492	159306	119350
$0.5WL + 0.5OF$	839052	159465	113754
$0.4WL + 0.6OF$	842840	153849	117805
$0.2WL + 0.8OF$	849358	159374	110485
$(1 - \alpha_T)WL + \alpha_TOF$	859994	156729	72723
<b>LkAhd</b>	881915	161172	71997
<b>QL</b>	840739	152345	72526
<b>LQ</b>	879860	160625	72593

TABLE XIII

COMPARISON BETWEEN DIFFERENT OBJECTIVES FOR CIRCUIT *avql*.

congestion-related objectives. This fact suggests that other congestion-related objectives are ill behaved. They are not better than the wirelength objective. However, we know that in practice the placement with minimal wirelength does not always satisfy the congestion constraint. Therefore, we need to find a new way to reduce the congestion more effectively.

## VI. POST PROCESSING TO MINIMIZE CONGESTION

We propose a two stage process to reduce the congestion in a layout. In the first stage, we use the wirelength as the objective to minimize the average congestion. After the first stage is done, we can perform post processing to further reduce the congestion. In the post processing stage, we use the overflow with look-ahead cost as the objective to minimize. In the post processing stage, we propose three types of algorithms:

1. Greedy cell-centric algorithm: This algorithm randomly moves cells around and only accepts moves which result in a reduction in the congestion overflow.

2. Flow-based cell-centric algorithm: This algorithm uses a flow-based approach to move multiple cells simultaneously.
3. Net-centric algorithm: This algorithm first sorts all the nets based on their contribution of congestion. Then it tries to move the nets one by one to reduce congestion.

The greedy cell-centric algorithm is straightforward and easy to implement. We evaluate moving a cell or exchanging two cells using the modified overflow objective. Then we make this move or exchange if and only if it can give us a lower objective cost value. This algorithm is quite simple and serves as a reference point to other algorithms.

The cell-centric random moving strategy proposed above is very greedy. It does not have the ability to know where the congestion is and how to reduce it. To improve, we propose a net moving strategy which can identify the highly congested spot and try to move nets out of this spot. The greedy feature of the above algorithm makes it easy to get stuck into a local minimum. To solve this problem, we propose a multiple cell moving strategy based on a network flow method. We try to find better locations for cells to reduce congestion. This can be viewed as a transportation problem. In the corresponding transportation problem, the source of the transportation is all the cells and the destination is all the global bins. A transportation cost is associated with a cell move. We then simultaneously transport the cells to new locations that minimize the transportation cost. Since the congestion cost is not linear, we do not allow more than one cell moved in/out any global bin in each iteration. At each iteration, the transportation problem can be transformed into a minimal-cost maximum flow problem [6] on a network as shown in Fig. 8. This network consists of a source node  $S$  supplying cells, a set of cell nodes  $\mu$ , a set of location nodes  $\lambda$ , and a destination node  $D$ . The capacities of arcs between node  $S$  and cell nodes are 1 implying that a cell can be moved only once in one iteration. Suppose each location can hold  $s_\lambda$  cells, the capacity of arc leading from a location node to node  $D$  is set to  $s_\lambda$ . The cost of moving a cell  $\mu$  to location  $\lambda$  is  $c_{\mu\lambda}$ , where  $c_{\mu\lambda}$  is the change in the objective when moving cell  $\mu$  to location  $\lambda$ . By using the flow augmentation method [3], [6], we can get a new location assignment of cells with minimum total transportation cost at each iteration.

Given a placement, we first route all nets. Then we assign a weight to each net. The weight of a net is equal to the number of overflowed global edges the net crosses. We sort the nets in descending order according to their weights. The net with the greatest weight is the one which contributes the most to the total overflow. Thus moving this net will most likely to help reducing the congestion. In order to move a net, we consider moving all cells connected to the net. The destination of the move could be any global bin. Thus we look at all the cells connected to the net and move a cell to a new position which can result in a reduction in the congestion overflow. After all the nets have been tried, we will update the net weights according to the new global routing information. We will

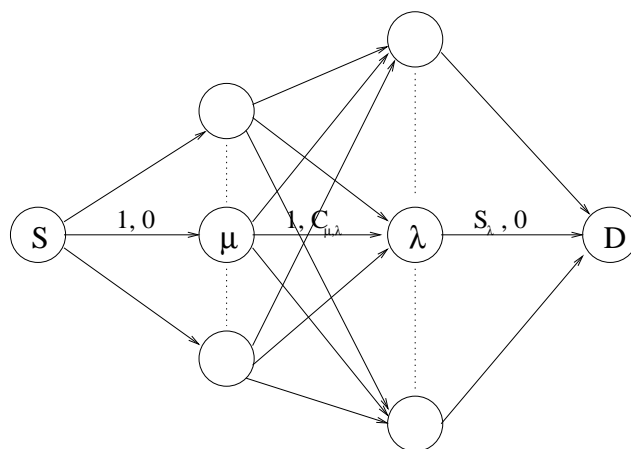


Fig. 8. Transportation network.

repeat the above procedure until the congestion overflow cannot be further reduced. Since congestion is essentially produced by nets, moving nets out of the congested region makes more sense than blindly moving single cells.

We run simulated annealing using the wirelength objective in the first stage. The output placement from the first stage will be the input to the post processing stage. Table XIV shows the results from the post processing stage. The *before PP* column in the table is the results before the post processing stage. The percentage improvement column is the improvement of using the post processing stage compared to the results before post processing. The post processing stage can significantly reduce the congestion cost if the input placement is good. We get an average 36.9% improvement compared to the congestion results before post processing. Among all the congestion reduction methods studied in this paper, this post processing method using the net-centric algorithm produces the best results.

Test Case	before PP	cell-centric	flow-based	net-centric	%imp.net-centric.vs.beforePP
highway2	12	7	7	<b>7</b>	41.7%
fract	16	14	14	<b>14</b>	12.5%
Primary1	34	9	17	<b>4</b>	88.2%
Primary2	151	56	65	<b>49</b>	67.5%
struct	88	52	39	<b>47</b>	46.5%
biomed	3011	2646	*	<b>2610</b>	12.1%
avqs	159	124	*	<b>116</b>	27.0%
avql	802	753	*	<b>747</b>	6.9%
ave.					36.9%

TABLE XIV

POST PROCESSING RESULTS USING DIFFERENT ALGORITHMS.

(\* out of memory)



## VII. FROM GLOBAL PLACEMENT TO DETAILED PLACEMENT

In this paper, congestion minimization is done in the global placement stage. In our global placement context, cells are located at the centers of global bins and the congestion of the chip is estimated based on that. However, in final placement, cells should be placed in a non-overlapping fashion. The congestion estimated from this non-overlapping placement will be different than the congestion estimated from our global placement. Given a global placement, we can construct a corresponding non-overlapping placement by spreading the cells within each global bin. This spreading procedure is usually called “detailed placement” and it may involve low temperature annealing followed by some simple (e.g., greedy) optimization procedures to determine the orders of cells within one global bin. In this section, we will show that the congestion estimated from the global placement is correlated with the congestion estimated from the corresponding detailed placement.

We will start with two global placements, one wirelength optimized placement ( $WL_g$ ) obtained by a traditional placement method and one congestion optimized placement ( $CON_g$ ) obtained by using the post-processing stage. Then we use a detailed placement algorithm to transfer global placements to detailed placements ( $WL_g \rightarrow WL_d$ ,  $CON_g \rightarrow CON_d$ ). We evaluate the overflows of these four placements. Since  $CON_g$  is the congestion minimized placement, it is expected that the overflow value of  $CON_g$  is less than the overflow of  $WL_g$ . Then if the overflow of  $CON_d$  is also less than the overflow of  $WL_d$ , we say that the overflow of global placements correlate with the overflow of detailed placements. All overflow values are estimated using the same global bin grids. We use TimberWolf.1.4.1 as our detailed placement algorithm (spreading procedure) to transfer a global placement to a detailed placement. TimberWolf.1.4.1 can read in an existing global placement and spread the cells. It also does some local wirelength optimization.

Table XV shows the results of this correlation test. The results show that the overflows of global and detailed placement correlate each other very well. Thus a less congested global placement will most likely produce a less congested detailed placement.

In the global bin context, congestion is ignored inside a global bin based on the definition of overflow. Thus when a global bin contains a large number of cells, the congestion estimation on this bin grids may not be accurate. Thus we should use finer global bin grids to estimate or optimize congestion. In this paper, which global bin grids to use is not the question of interest. What we have shown here is how to effectively reduce the congestion with given global bin grids. The post-processing method we proposed here should be valid for different global bin sizes.

## VIII. CONCLUSION

In this paper, we studied the behavior of congestion minimization in placement. As shown both by theoretical and

TestCase	$WL_g$	$CON_g$	$WL_d$	$CON_d$
highway2	12	8	18	13
fract	16	14	24	23
Primary1	140	125	151	141
Primary2	710	586	917	867
struct	150	110	261	227
biomed	667	1115	605	1084
avqs	180	149	258	214
avql	898	791	1032	909

TABLE XV  
CORRELATION TEST BETWEEN GLOBAL PLACEMENT AND DETAILED PLACEMENT.

experimental results, the congestion cost is a poorly behaved function. Our theoretical analysis showed that there are some correlations between wirelength and congestion in a placement. Specifically, the total wirelength is equal to the total routing demand of a global placement. Therefore, minimizing wirelength is helpful in minimizing congestion globally. In order to understand the problem of minimizing congestion in placement, we tested seven different congestion related objectives. We proposed a post processing stage with a very effective net-centric algorithm to reduce congestion in a layout. To summarize our results:

1. Wirelength minimization can minimize congestion globally. A post processing congestion minimization following wirelength minimization works the best for reducing congestion in placement.
2. We tested a number of congestion-related cost functions including a hybrid length plus congestion (commonly believed to be very effective). Experiments prove that they do not work very well.
3. Net-centric post-processing techniques are very effective to minimize congestion.
4. Congestion at the global placement level, correlates well with congestion of detailed placement.

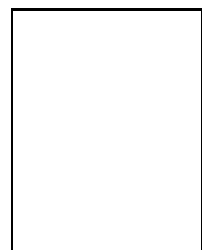
## ACKNOWLEDGMENTS

This work was support in part by NSF grant MIP-9527389.

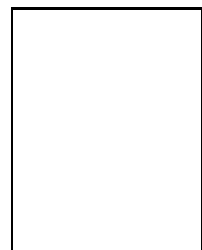
## REFERENCES

- [1] A. E. Dunlop and B. W. Kernighan, *A Procedure for Placement of Standard Cell VLSI Circuits*, IEEE Transactions on Computer Aided Design, 4(1): 92-98, January 1985.
- [2] H. Eisenmann and F. M. Johannes, *Generic Global Placement and Floorplanning*, In Design Automation Conference, pages 269-274, IEEE/ACM, 1998.
- [3] L.R. Ford and D.R. Fulkerson, *Flows in Network*, Princeton, NJ, 1962.
- [4] D. Huang and A. B. Kahng, *Partitioning-based Standard-cell Global Placement with an Exact Objective*, In International Symposium on Physical Design, pages 18-25, ACM, April 1997.
- [5] J. M. Kleinhans, G. Sigl, F. M. Johannes and K. J. Antreich, *GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization*, IEEE Transactions on Computer Aided Design, 10(3): 365-372, 1991.
- [6] T. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, John Wiley & Sons, 1990.

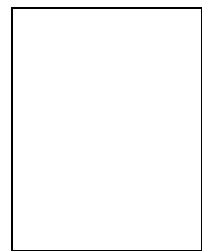
- [7] G. Meixner and U. Lauther, *Congestion-Driven Placement Using a New Multi-Partitioning Heuristic*, In International Conference on Computer-Aided Design, pages 332-335, IEEE/ACM, November 1990.
- [8] P. N. Parakh, R. B. Brown and K. A. Sakallah, *Congestion Driven Quadratic Placement*, In Design Automation Conference, pages 275-278, IEEE/ACM, 1998.
- [9] Saab96, *A Fast Clustering-based Min-cut Placement Algorithm with Simulated-annealing Performance*, VLSI Design: An International Journal of Custom-Chip Design, Simulation, and Testing, 5(1): 37-48, 1996.
- [10] M. Sarrafzadeh and M. Wang, *NRG: Global and Detailed Placement*, In International Conference on Computer-Aided Design, pages 532-537,
- [11] M. Sarrafzadeh and M. Wang, *Interaction Among Cost Functions in Placement*, In International Conference on VLSI and CAD, 1999.
- [12] C. Sechen, *VLSI Placement and Global Routing Using Simulated Annealing*, Kluwer, B. V., Deventer, The Netherlands, 1988.
- [13] K. Shahookar and P. Mazumder, *VLSI Cell Placement Techniques*, ACM Computing Surveys, 23(2): 143-220, June 1991.
- [14] P. R. Suaris and G. Kedem, *Quadrisection: A New Approach to Standard Cell Layout*, In Design Automation Conference, pages 474-477, IEEE/ACM, 1987.
- [15] R. S. Tsay, E. S. Kuh and C. P. Hsu, *PROUD: a Sea-of-Gates Placement Algorithm*, IEEE Design and Test of Computers, pages 44-56, December 1988.
- [16] M. Wang and M. Sarrafzadeh, *On Behavior of Congestion Minimization During Placement*, In International Symposium on Physical Design, pages 145-150, ACM, April 1990.



**Maogang Wang** received his B.S. degree in 1994 from the University of Science and Technology of China at Hefei, China. He received his M.S. degree in Physics and Computer Engineering in 1996 and 1998 respectively from Northwestern University. From 1996 to 2000 he worked with Professor Majid Sarrafzadeh as a Ph.D. student in NuCAD lab of the department of Electrical and Computer Engineering at Northwestern University. His research interests lie in the area of physical layout in VLSI CAD. He received his Ph.D. degree in May 2000.



**Xiaojian Yang** received the B.S. degree in computer science from Tsinghua University, China in 1994, and M.S. degree from Chinese Academy of Sciences in 1997. He is currently a Ph.D. student in Professor Majid Sarrafzadeh's NuCAD lab in the department of Electrical and Computer Engineering at Northwestern University. His research interests include logic synthesis and physical design, with an emphasis on wirelength, congestion and timing issues in deep sub-micron placement.



**Majid Sarrafzadeh** received his B.S., M.S. and Ph.D. in 1982, 1984, and 1987 respectively from the University of Illinois at Urbana-Champaign in Electrical and Computer Engineering. He joined Northwestern University as an Assistant Professor in 1987. Since 1997 he has been a Professor of Electrical Engineering and Computer Science at Northwestern University. His research interests lie in the area of VLSI CAD, design and analysis of algorithms and VLSI architecture. Dr. Sarrafzadeh is a

Fellow of IEEE for his contribution to "Theory and Practice of VLSI

Design". He received an NSF Engineering Initiation award, two distinguished paper awards in ICCAD, and the best paper award for physical design in DAC for his work in the area of High-Speed VLSI Clock Design. He has served on the technical program committee of numerous conferences in the area of VLSI Design and CAD, including ICCAD, EDAC and ISCAS. He has served as committee chairs of a number of these conferences, including International Conference on CAD and International Symposium on Physical Design.