

A New Congestion-Driven Placement Algorithm Based on Cell Inflation*

HOU Wen-ting¹, YU Hong¹, HONG Xian-long¹, CAI Yi-ci¹, WU Wei-min¹ and GU Jun²

(1 Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

(2 Department of Computer Science and Technology, University of Hong Kong, Hong Kong, China)

Abstract: A new congestion-driven placement Algorithm is described based on the cell inflation. In this approach, the methods of probability-estimation and star-model are used to evaluate the routing of nets. Global placement can be done by using the algorithm of global optimization and slicing partitioning. The denotation of virtual area of cell is given to indicate not only the area of cell but also the routing demand. The virtual area of a cell is got by using the strategy of cell inflation, with which in the slicing partitioning, the routing congestion is eliminated. Further reduction in congestion is achieved by cell moving. The algorithm has been tested on a set of sample circuits from American companies, with great improvement in routability having been obtained.

Key words: congestion; probability-estimation; cell inflation; cell moving

EEACC: 2570 **CCACC:** 7410D

CLC number: TN405.97 **Document code:** A **Article ID:** 0253-4177(2001)03-0275-08

1 Introduction

The problem of introducing performance constraints into the automatic layout of integrated circuits has been addressed in the past. With the advent of over-the-cell routing, the goal of every place and route methodology has become to utilize an area to prevent the routes from spilling into the channels. It is this overflow of routes that accounts for an increase in areas. The multiple routing layers have enough routing resources to route the most wires as long as there are not too many wires congested in the same region. Yet net list synthesis tools tend to generate net lists with high pin-count nets and poor porosity library cells, which often result in the local routing congestion^[1].

Typical placement objectives are either to reduce the net-cut costs or to minimize the wire

length. Because of its constructive nature, min-cut based strategies can minimize the number of net crossings but fail to uniformly distribute them^[2]. Congestion-driven placement based on multi-partitioning has been proposed in Ref. [3]. The actual congestion cost calculated from the pre-computed Steiner trees are used to minimize the congestion of the chip, however, the number of partitions is limited due to the excessive computational load^[4]. The minimal wire length as the metric to guide placement, has succeeded in the achievement of good placement. However, it only indirectly models the congestion and the behavior of the router. Reducing the global wire length is helpful to reduce the wiring demand globally, but can not prevent the existing local congested spots. Therefore, traditional placement schemes that are based solely on the wire length minimization^[5-7] cannot adequately account for the congestion.

Recently, some congestion-driven placement

* Project Supported by National Natural Science Foundation of China(No. 69776027) and 973 National Key Project (No. G1998030411).

methods are reported, such as the one proposed by Parakh, which uses the strategy of “region expanding” to eliminate the routing congestion^[8] and leads to the congestion oscillation in two or more regions easily. As for the method presented by Wang^[9], besides the conventional placement algorithms, a global router is run to evaluate the congestion in the placement stage. And then, a flow-based cell-centric algorithm is carried out to move the cells so as to minimize the congestion. However, this method depends on the router heavily.

In this paper, a new congestion-driven placement algorithm based on cell inflation is proposed, in which, the congestion keeps checking and eliminating while we doing the placement. To check the routing congestion, the method of “resource estimation” is used instead of global routing, and the result of congestion is more common and not router-dependent. “Cell inflation” is used to alleviate and erase the congestion. All of above are done self-adaptively in this approach, so that not only the “congestion oscillation” can be avoided, but also it is of small loss in total wire length. Finally, “cell moving” is described to refine our solution further.

2 Congestion Analysis

2.1 Bin Structure

A poorly placed layout might be un-routable due to the limited routing resources. In order to ensure the maximum routability in the layout, the congestion should be minimized in the placement stage. The followings are some denotations.

Global bin structure: A given chip is partitioned into a set of rectilinear regions, each of which is called a bin, as is the same as Reference [4]. The congestion is defined based on the bin structure. For example, in Figure 1, we have $4 \times 4 = 16$ bins. For a certain bin, when its routing demand exceeds its routing resource supply, it is called a congested-bin. The number of bins must

be set properly; if it is too small, it will result in the poor accuracy; on the contrary, it will increase the computing complexity.

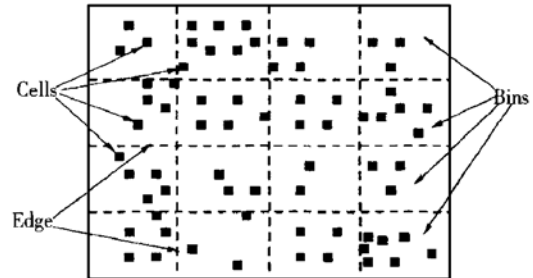


FIG. 1 Bin Structure

D_{ij} (demand): In bin(i, j), D_{ij} is the number of routes, which are across or inside the bin. Since there are horizontal and vertical routes in each bin, there exist two D_{ij} in each direction.

S_{ij} (supply): In bin(i, j), S_{ij} is the number of tracks, which are supplied by the route layer. There are both S_{ij} in vertical and horizontal directions, too. Existent wiring of power and clocking nets, and standard cells are considered obstacles to the routing.

2.2 Wiring Modeling

If we estimate the routing congestion via running a global router, the problem of “router-dependent” will appear. To avoid it, the method of routing-estimation is adopted. In the course of placement, a quick estimation is made for each net using this method, with the congested bins being detected. After that, the “cell inflation” is used to eliminate the congestion.

The routing estimation method applies to the standard cell layout with two or more routing layers. From the current cell positions and bin structures, the routing estimation of each net across the chip plan is obtained in both horizontal and vertical directions, as well as the information on the congested bin.

Since the routing estimation is done during the placement, we do not know the real shape of each

net. In fact, even a two-pin net can be routed in many forms, so the follows are criteria to build up an estimation model:

- 1) No detours exist when we estimate a net, and the net must be routed within its bounding-box.
- 2) The estimation should be appropriate.
- 3) No complicated computing during the routing estimation.

In practice, we adopt “star-model” to estimate the routing of each net. Therefore, it is necessary to find the center of the net according to formula(1):

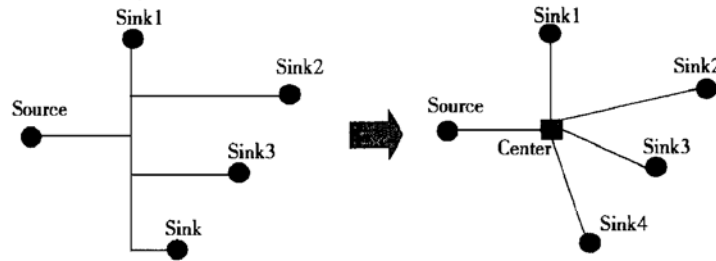


FIG. 2 Star-Model

To estimate the routing of nets, we have to follow the rules below: a) all the two-pin nets should route without detours and within its bounding-box, and any two-pin net can switch direction once at most; b) the star-mode net can be transformed into a rectilinear routing tree based on “Manhattan Routing”. As shown in Fig. 3, when we estimate the route that connects to Sink2 and the Center, there are two possible routing paths: *A* and *B*. No one knows which path the following router will choose after placement, so we assume that the probabilities of the following router’s choosing path *A* or *B* are equal, i. e. this route will go along path *A* and path *B* 0.5 track for each. Obviously, if the route has the same *x* or *y* coordinate, there would be only one possible path. As shown in Fig. 3, when we route between Source and Center, there is only one path, so that the routing estimation is 1.0 track. According to this method, the result of routing estimation is obtained and shown in Fig. 4. Please note that in

$$x_m = \sum_{i \in n} x_i / |n|, \quad y_m = \sum_{i \in n} y_i / |n| \quad (1)$$

In order to optimize the routing and reduce the number of vias, we adjust x_m and y_m to the nearest coordinates of one of the nodes that connects to the net, and then connect each node of this net with the center. Figure 2 is a five-pin net. First, the net is transformed into star-mode just like the one on the right. Then we compute and adjust the coordinates of the net center, where the *x* coordinate of it equals to that of node Sink1, while *y* coordinate equals to that of node Source.

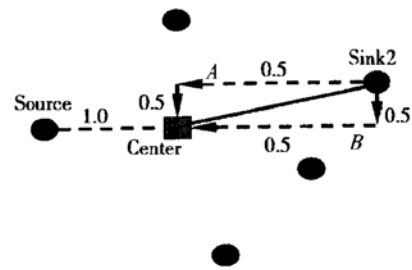


FIG. 3 Routing Estimation

some segments, the estimated number of the required tracks is more than 1.0. It is because there are some duplicated nets in these segments. As it is not consistent with the real routing, we modify

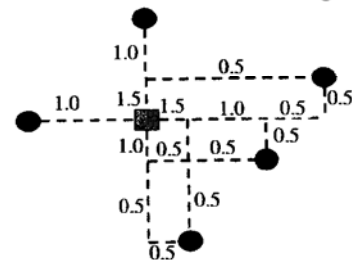


FIG. 4 Routing Estimation of Five-Pin

them into a single net. In addition, the number of tracks required in the bins, where the net passes, should be recorded.

In fact, the “star-model” is very similar to the “minimum steiner-tree” model, and the net center can be compared to the steiner point. According to our experiments, “star-model” is very close to the real routing in practice.

2.3 Routing Resource Supply

The information on the routing resource supply includes the width of each metal layer, the pitch, positions of pins in cells, obstacles and so on.

2.3.1 Existent wiring

An existing wire in layer m of the power, ground or clocking nets in a region (i, j) , shown in Fig. 5(a), decreases the routing resource supply by a/w , where w is in the unit of routing tracks. Since these wires are fixed throughout the placement, no updating is required during the run time of placement once the resource supply is reduced.

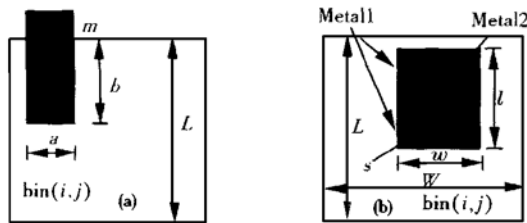


FIG. 5 (a) Existing Wiring; (b) Cell Obstacle

2.3.2 Cell Obstacles

A cell normally consists of one blockage in the first metal layer, which is as large as the cell outline, several blockages in the second metal layer and some pins in either the first or the second metal layer. We assume that metal1 and metal2 are horizontal routing and vertical one, respectively. Therefore, a cell placed in bin (i, j) decreases the routing resource supply by:

$$\Delta_{\text{metal1}} S_{ij} = T_1 \times \frac{l}{L} \quad (2)$$

$$\Delta_{\text{metal2}} S_{ij} = T_2 \times \frac{s}{W} \quad (3)$$

where l is the length of the cell indicated in Fig. 5 (b), L and W are those of the bins, T_1 and T_2 are the numbers of tracks in the first two metal layers within a bin, respectively. Let s be the width of pins in a cell. Note that though it is in the first metal layer that the pins might be, they are still considered obstacles in the second metal layer to other nets because of their accessibility. The decrement in both equations (2) and (3) can be pre-processed before the global optimization.

2.4 Cell Inflation

Let's introduce some more denotations again.

Real Area of cell: The real area of a cell is the product of the length and the height of a cell, which is the actual size of the cell and can be used in the accurate computation, such as in the first step in global optimization to get the global optimized position of the cell, in the routing estimation, etc.

Virtual Area of cell: The virtual area of a cell is the area of the cell after inflation. The value reveals not only the size of the cell but also the routing demand of the bin, where the cell locates currently. The virtual area can be used to eliminate the congestion, such as in the latter step in the global optimization to decrease the routing congestion of the bins. The value of the virtual area is a multiple of the real area, and the ratio can be determined by the degree of the congestion of the bin.

The method of cell inflation is presented as follows.

After the routing estimation and detection of the congested bins, in which the real area of the cell is considered accurate, what we should do is to inflate the areas of the cells located in the congested region. Supposing bin A is a congested one and the area of cells located in bin A have the same ratio, we can get the virtual area of the cells, which will be used during the implementation of the next partition loop. Due to the rule of area-balance, some cells located in bin A are forced to move to

other bins, as results in a decrease in the number of cells located in bin *A*. Generally speaking, with the decrease in the number of cells, the number of nets through bin *A* will decrease accordingly. As a result, the routing resource supply in bin *A* will increase, while the routing demand will decrease. Therefore, the congestion in bin *A* will be alleviated or eliminated. When the inflated cell moves to an uncongested bin completely, it will recover to the original area.

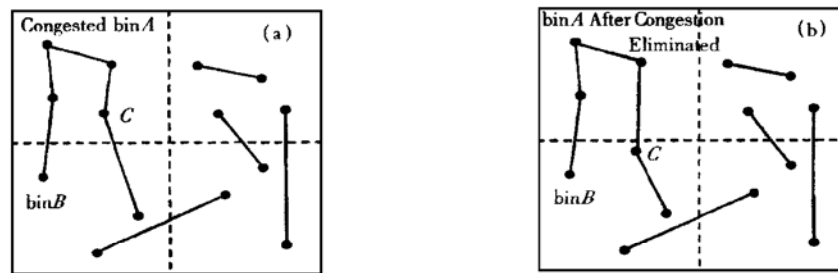


FIG. 6 (a) Routing Congested in bin *A*; (b) Routing Congestion Eliminated

2.5 Cell Moving

After the global placement, the cell moving is implemented to erase the routing congestion further, which is a desired method to decrease the congestion. For instance, we now have a most congested bin *S* and a most un-congested bin *T*, so we draw a line from *S* to *T*, and then use a step line *SABCD...T* to fit it. Next, some operations are carried out along the step line. The cells in bin *S* to bin *S* and *A* are mapped according to the coordinates, as well as those in *A* originally and the newcomer ones from bin *S* to bin *A* and *B*. The same operation will not be finished until we reach bin *T*. After that, we do the congestion detection and find another most congested bin and the most un-congested bin. We then repeat the above cell moving for several times.

The experimental results show that the cell moving can decrease the congestion considerably. The total wire length can be increased during the cell moving. Since the cells are merely moved to

As shown in Fig. 6(a), there are four cells and five nets in the left-up bin *A*, so it is congested. We expand the areas of the four cells located in bin *A*. After next partition loop, due to the area-balance, cell *C* is forced to move to bin *B* by using the virtual area of cell. At this time, there are only three cells and four nets in bin *A*, so the routing resource supply increases while the routing demand decrease. The congestion in bin *A* is then eliminated.

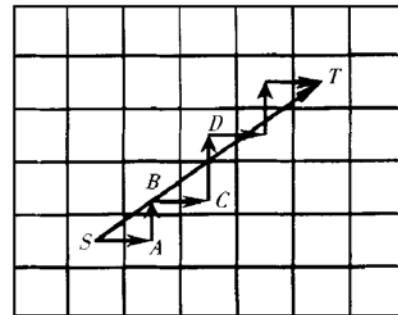


FIG. 7 Cell Moving

the adjacent bins, the increment is limited. The ratio of nets routed is overwhelming for a router. Because the moving can not only enhance the routability, but also make the route easier and decrease the detour, it is fairly valuable.

3 Main Algorithm Description

The information on the circuit is read through LEF/DEF format. An $m \times n$ bin structure is built up.

3.1 Global Optimization and Slicing Partition

The main loop of the global placement is composed of global optimization, congestion detection, congestion elimination and slicing partition.

In global optimization, a mathematical programming problem is derived and then solved. The objective function is based on the nets' quadratic wire length model. For a net n , its length is computed according to the following equation:

$$L_n = \sum_{i,j \in n} [(x_i - x_j)^2 + (y_i - y_j)^2]$$

Since the number of cells in a net is different from each other, we assign a weight w_n to net n . Therefore, the objective is to minimize the weighted sum of the quadratic wire length of all nets:

$$\Phi = \sum_{n \in N} L_n w_n$$

At the l th partition level, the placement plane is divided into 2^l regions, each containing a subset of cells. If r is a region, we use τ to denote the set of cells contained in r , and (μ_r, ν_r) to denote the coordinates of the center of region r . Thus for each region r , we have two constraints on the global placement:

$$\frac{\sum_{i \in \tau_r} x_i}{|\tau_r|} = \mu_r \quad \frac{\sum_{i \in \tau_r} y_i}{|\tau_r|} = \nu_r$$

Combining the objective function with the constraints, we get a constrained quadratic programming problem:

$$LQP: \min\{\Phi(x) = \frac{1}{2}X^T QX + b_x^T X \mid A^{(l)}X = u^{(l)}\}$$

With the Lagrange Relaxation Method, this problem can be solved. According to Reference [10], the global optima can be obtained by:

$$x = Q^{-1}b_x \quad y = Q^{-1}b_y.$$

where the vectors x and y denote the coordinates of the movable cells to be placed. We solve this problem by changing the format of above formula and making use of the iterative solution method——conjugate-gradient method (CG). To improve the condition number of matrix Q , we use the preconditioned conjugate-gradient method (PCG) in prac-

tice, which can result in an effective solution.

First, the global optimization and partition are achieved on the chip planes, from level 0 to the partition-level limit N . In this way, we get the global optimization for each cell. Then, we detect the routing congestion and find the congested bins. During above processes, the real area of the cell as the accuracy, is used to get the optimized optimization. Next, with this method, we expand the areas of cells to get the virtual areas of cells and adjust the positions of cells to eliminate the congestion. For a better result, we withdraw the partition level to level $(N - 6)$ and restart our global optimization and partition, using the virtual areas of cells. Because of the rule of partition, part of the cells in the congested bins move out of the bins automatically and naturally. When reaching the partition limit N again, we do the congestion detection, expand the areas of cells to update the virtual areas of cells, and withdraw the partition level again. After repeat it several times, we draw a conclusion, with the loop flow shown in Fig. 8.

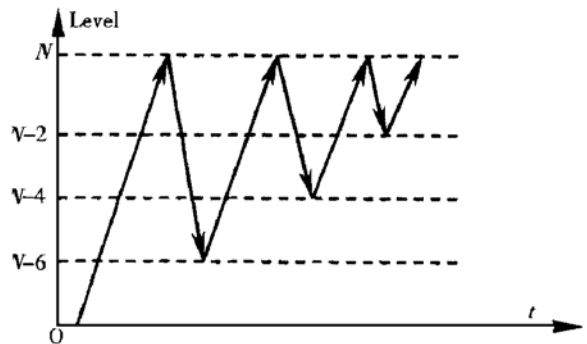


FIG. 8 Partition Loop Flow

The algorithm can decrease the maximum congestion markedly without any congestion vibration. With this method, the total wire length is increased. However, because the position of cells is adjusted naturally, it only increases a little. Though we spend more time in withdrawing the partition level repeatedly during this algorithm, it is still timesaving and effective.

3.2 Post-Processing

After global placement, the cell moving is

performed to erase the routing congestion further, as described in section 2.4. Finally, a final placement, namely FAME, is used to assign the cells into the slots and refine the solution obtained. For more about FAME, please refer to Reference[11]. In the last two processes, we both use the real areas of cells, so the inflated method will not affect the final placement. The main design flowchart of our approach is shown in Fig. 9.

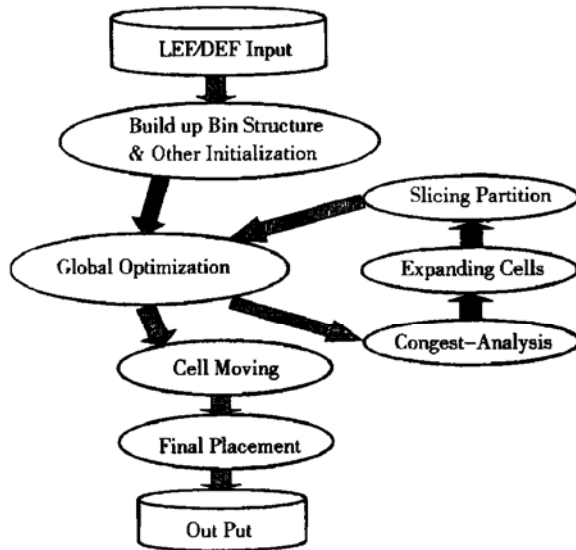


FIG. 9 Main Design Flow chart

4 Experimental Results

New congestion-driven placement algorithm

has been implemented on ULTRA-SPARC workstations with C language. To verify this approach, we have tested it with a set of sample circuits from American company. The characteristics of the circuits are listed in Table 1. As for the approach without congestion constraints, the results, which are obtained by using conventional placement (CP) and congestion-driven placement (CDP), are summarized in Table 2 and 3. X-Max is the number of track shortage in the most congested bin in x direction, while Y-Max is that in y direction. According to the results, it is shown that our approach can effectively cut down the maximum congestion at about 30–40 percent. Although the total wire-length increases slightly, it is less important than routability by now, especially in the cases of deep submicron technology. Even though the run time increases as twice as before, our algorithm still runs fairly fast. It takes about one hour for the circuit U28070 that consists of more than 160 thousand cells.

Table 1 Characteristics of Sample Circuits

# Circuits	# Pin	# Cell	# Net	# Row	# Layer
Table	142	4596	4674	64	4
Design	1028	44283	44943	164	5
DM	1607	82924	83990	172	5
U05614	388	32112	36452	186	3
U11228	836	64224	72966	263	3
U28070	1604	160560	181932	417	3

Table 2 Comparison Between CP and CDP

Circuits	CP				CDP			
	Wire/ μm	R-time/s	X-Max	Y-Max	Wire/ μm	R-time/s	X-Max	Y-Max
Table	33643475	73.6	90	16.5	33934358	212.7	58.5	4.5
Design	50503240	542.4	193	106.5	50772567	1187.9	164	85
DM	1.17×10^8	766.6	1014	1011	1.18×10^8	1703.6	680	720
U05614	26148356	543.9	553	455	27246965	1255	410	400
U11228	62317492	938.1	1287	1078	64068224	1777.6	699.5	588.5
U28070	2.16×10^8	1714	4173	4032	2.26×10^8	3637.4	2195	1728

Table 3 Ratio of CDP vs CP

Circuits	Wire	R-Time	X-Max	Y-Max
	Increase	Increase	Decrease	Decrease
Table	0.86	188.99	35.00	72.73
Design	0.53	119.01	15.03	20.19
DM	0.63	122.23	32.94	28.78
U05614	4.20	130.74	25.86	12.09
U11228	2.81	89.49	45.65	45.41
U28070	4.96	112.22	47.40	57.14

5 Conclusion

In this paper, we have presented a new congestion-driven placement based on cell inflation, in which the method of probability-estimation is used to evaluate the routing of nets. Using the schemes

of cell inflation, we eliminate the routing congestion. Further reduction in congestion has been obtained via cell moving. In addition, the well-known quadratic placement is incorporated with the slicing partitioning strategy in this approach. The experimental results prove this method effective and efficient.

References

- [1] Chih-liang Eric Cheng, RISA: Accurate and Efficient Placement Routability Modeling, IEEE/ACM Proceedings of International Conference on CAD, 1994.
- [2] Y. Saab, A Fast Clustering-based Min-cut Placement Algorithm with Simulated-annealing Performance, VLSI Design: An International Journal of Custom-Chip Design, Simulation, and Testing, 1996, 5(1): 37—48.
- [3] S. Mayrhofer and U. Lauther, Congestion-Driven Placement Using a New Multi-Partitioning Heuristic, International Conference on Computer-Aided Design, IEEE/ACM, November, 1990, 332—335.
- [4] Maogang Wang and Majid Sarrafzadeh, On the Behavior of Congestion Minimization During Placement, Proceedings of International Symposium on Physical Design, 1998, 145—150.
- [5] J. M. Kleinhans, G. Sigl, F. M. Johannes and K. J. Antreich, GORDIAN: VLSI Placement by Quadratic Programming and Slicing Optimization, IEEE Transaction on Computer Aided Design, 1991, 10(3): 356—365.
- [6] H. Eisenmann and F. M. Johannes, Generic Global Placement and Floorplanning, IEEE/ACM Design Automation Conference, 1998, 269—274.
- [7] Kong Tian-ming, Hong Xian-long and Qiao Chang-ge, VEAP: Global Optimization Based Efficient Algorithm for VLSI Placement, Chinese Journal of Semiconductors, 1997, 18: 692.
- [8] P. N. Parakh, R. B. Brown, K. A. Sakallah, Congestion Driven Quadratic Placement, Proceedings of IEEE/ACM Design Automation Conference, 1998, 275—278.
- [9] Maogang Wang and Majid Sarrafzadeh, Modeling and Minimization of Routing Congestion, Proc. of ASP-DAC2000, 185—190, 2000, Japan.
- [10] Hong Yu, Xianlong Hong, Changge Qiao and Yici Cai, CASH: A Novel Quadratic Placement Algorithm for Very Large Standard Cell Layout Design Based on Clustering, Proceedings of the 5th International Conference on Solid-State and Integrated Circuit Technology, 1998, 496—501.
- [11] B. Yao, W. T. Hou, X. L. Hong and Y. C. Cai, FAME: A Fast Detailed Placement Algorithm for Standard-Cell Layout Based on Mixed Mincut and Enumeration, Chinese Journal of Semiconductors, 2000, 21: 744.

一种新的基于单元扩大的拥挤度驱动的布局算法*

侯文婷¹ 于泓¹ 洪先龙¹ 蔡懿慈¹ 吴为民¹ 顾钧²

(1 清华大学计算机科学与技术系, 北京 100084)

(2 香港大学计算机科学与技术系, 香港)

摘要: 描述了一种新的基于单元扩大的拥挤度驱动的布局算法. 这个方法用概率估计模型和星型模型来评价线网的走线. 使用全局优化和划分交替的算法来进行总体布局. 提出了单元的虚拟面积的概念, 单元的虚拟面积不仅体现了单元的面积, 而且指出了对布线资源的需求. 单元的虚拟面积可以由单元的扩大策略来得到. 把单元的虚拟面积用到划分过程中, 从而减小拥挤度. 并且使用了单元移动的策略来进一步减小走线的拥挤. 用来自美国公司的一些例子测试了这个算法, 结果显示布局的结果在可布性方面有了很大的提高.

关键词: 拥挤; 概率估计; 单元扩大; 单元移动

EEACC: 2570 CCACC: 7410D

中图分类号: TN405.97 文献标识码: A 文章编号: 0253-4177(2001)03-0275-08

* 国家自然科学基金资助项目和 973 国家重点项目

2000-09-26 收到, 2000-12-06 定稿