Taylor & Francis
Taylor & Francis Group

IFDR: AN EFFICIENT ITERATIVE OPTIMIZATION ALGORITHM FOR STANDARD CELL PLACEMENT

FENG CHENG* and JUNFA MAO

Department of Electronic Engineering, Shanghai Jiaotong University, Shanghai 200030, People's Republic of China

In the automatic placement of integrated circuits, the force directed relaxation (FDR) method [Goto, S. (1981). An efficient algorithm for the two-dimensional placement problem in electrical circuit layout. *IEEE Trans. on Circuits and Systems*, **CAS-28**(1), 12–18] is a good iterative optimization algorithm. In this article, an improved force directed relaxation (IFDR) method for standard cell placement is presented, which provides a more flexible and efficient cell location adjustment scheme and a more extensive searching scale for better iterative placement optimization than the FDR method. A new heuristic algorithm based on local optimization is combined with the IFDR method to improve the placement. Experiments on the Microelectronics Center of North Carolina (MCNC) standard cell benchmarks [http://www.cbl.ncsu.edu/pub/Benchmark_ dirs/Layout Synth92/] have been done, and the results show that total wire length is reduced up to 25% and by an average of 16% in comparison with that from the placement algorithm of TimberWolf 7.0.

Keywords: Standard cell placement; Force directed relaxation; Iterative optimization; Circuit CAD

1 INTRODUCTION

Automatic cell placement has always been an important step in the fast and efficient design of VLSI circuits. A common formulation of the placement objective is to minimize total wire length under the constraint that cells do not overlap each other. As the placement problem is well studied, a number of approaches have been established. Up to now, standard cell placement methods can be mainly classified into three categories. The first class of algorithms is based on iterative optimization with determining cell locations iteratively. Among them, the force directed relaxation (FDR) approach is typical, which can model the wire length, cell congestion, and timing in the same mathematical formulation to allow smooth and simultaneous optimization of design in terms of these three parameters. The second class of algorithms is based on a simulated annealing algorithm. This method can obtain cell placements by swapping positions of cells randomly, guided by a probabilistic acceptance function. A well-known example is TimberWolf 7.0 (TW) [1], which can get good solution quality, but is fairly slow. The third class of algorithms is based on partitioning methodology [2]. It has become attractive recently and has good time efficiency for VLSI circuit place-

* Corresponding author. Tel.: +86-21-62933001; Fax: +86-21-6293001; E-mail: alma_cf@sjtu.edu.cn

ment, but its lack of global perspective and its min-cut objective mismatching with the real objective of wire length optimization lead to an adverse impact on overall solution quality.

This article presents an improved force directed relaxation (IFDR) algorithm, which determines cell target locations iteratively and more efficiently than the FDR method. A heuristic algorithm based on local optimization with the property of fast convergence is developed and combined with the IFDR method for the placement. This scheme of placement not only saves a large amount of time spent for blind cell movements in simulated annealing (SA), but also is more powerful than the partitioning-based methods, which make irreversible decisions at early stages based on incomplete or inaccurate information. Experimental results show that the total wire length in the application circuits from IFRD is reduced by up to 25% and an average of 16% in comparison with that from the placement algorithm of TW.

2 METHODS OF FINDING CELL TARGET LOCATIONS

2.1 The Force Directed Relaxation Algorithm

The FDR algorithm proposed in Ref. [3] is an efficient approach to determine cell target locations. The main idea of the algorithm can be described as follows.

Assume there are m fixed points in the x-direction, with coordinates of $x_1, x_2, \ldots, x_m$, respectively, and assume x is also the coordinate of a movable point in the x-direction. In order to obtain the minimum value of $\sum_{i=1}^{m} \overline{xx_i}$, the best coordinate of the movable point is:

(i) between $[x_k, x_{k+1}]$, $k = m/2$, when k is even;
(ii) at point x_k, $k = (m+1)/2$, when k is odd.

Note that $\overline{xx_i}$ denotes the distance between the movable point and the fixed point with a coordinate of x_i.

The above algorithm is introduced to cell placement for finding cell target locations to obtain the minimal total wire length of a design. Since the FDR method transforms the two-dimension problem of placement into two one-dimension (x-direction and y-direction) placement problems, only the total wire length optimization problem in the x-direction is discussed in detail, while the optimization problem in the y-direction can be dealt with in a similar way. In this article, the half perimeter net bounding box estimation model is adopted, with the bounding box determined by the left corner of a cell.

In Ref. [3], a selected cell a, going to move, is regarded as the movable point in the FDR method, and the leftmost and rightmost points on every net connected to cell a are regarded as the fixed points. Here, cell a is excluded from the net when forming the net bounding box. Then the target location of cell a is calculated with the FDR method. However, the performance of excluding cell a when determining the fixed points in the FDR method is problematic. Take only one net connecting cell a, for example, the location of cell a on the net can be only in two cases: (i) in the middle of the bounding line and (ii) at the boundary of the bounding line. For the first case, the net length is a constant and has nothing to do with the location of cell a, so cell a should remain in the middle of the line. For the second case, the net length is only decided by the location of cell a and the other bounding point, thus, as long as cell a moves towards the other bounding point, the net length will be shortened. The movement into the middle of the line is not necessary, for such strict movement will extremely restrict the location adjustment of cell a and the affection causing a poor iterative result will be significant when there is more than one net connecting cell a. Although Ref. [3] tries to interchange cell a with more than two cells in the neighborhood of its target
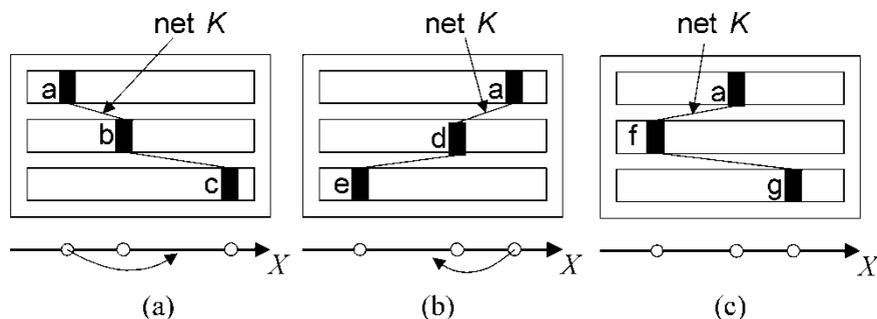
FIGURE 1 Three location cases of cell *a* on net *K*. In this figure, net *K* is connected with three cells (*a, b, c*; *a, d, e*; or *a, f, g*) and is projected to the *x*-direction.

location at the same time, the searching scale is still limited by the improper consideration of fixed points in the FDR method.

Therefore, the well-used FDR algorithm is improved for placement in this article. The improved algorithm applies three different ways successively to select the fixed points for finding cell target locations during the wire length optimization procedure. The advantage of our algorithm is that a flexible and efficient cell location adjustment scheme is applied so that a more extensive scale is searched for better iterative placement optimization than the FDR method.

2.2 The Improved Force Directed Relaxation Algorithm

The IFDR method applies three level selections of fixed points successively to three placement phases, which can be explained as follows.

The first level selection of fixed points is for the original large region adjustment of cells, which is described in Figure 1. There are three location cases of cell *a* on net *K* as shown in Figures 1(a)–(c). In Figures 1(a) or (b), the fixed point of net *K* should be point *c* or *e*. In these two cases, cell *a* is dragged into the middle of the net to minimize the wire length. In Figure 1(c), the fixed points of net *K* should be points *f* and *g* to keep cell *a* in the middle of the net and preserve the minimal wire length.

At the second level selection of fixed points, the target location of cell *a* is estimated first with the FDR method, then the exact fixed points are selected with the first level selection approach based on the estimated location of cell *a*. It is a relatively precise adjustment after the large region adjustment of cells.

The third level selection of fixed points is for the final small region adjustment of cells, that is, the fixed points should be *b, d, f*, and *g*, correspondingly. This level is just opposite to the first and second level selections and a compensation for them.

3 PLACEMENT ALGORITHM

3.1 Basic Algorithm

The initial placement in our placement algorithm is random, which means that all the cells are placed one by one according to the original netlist file. In our iterative optimization algorithm, each cell is chosen only once to move to its target location in a cell location adjustment cycle. During the placement optimization procedure, a heuristic algorithm based on local

optimization is used. This heuristic algorithm is an iteration of the process on a perturbed solution, that is, whenever a locally optimal placement is reached, the next cell location adjustment cycle should be accepted unconditionally. Based on the heuristic algorithm, we divide the whole placement optimization procedure into three phases. In the first phase, the first level selection of fixed points in the IFDR method is adopted to find cell target locations on a large scale. When the perturbed solution cannot keep quick descent tendency any more, the optimization procedure turns to the second phase with the second level selection. This procedure goes on until the third level selection in the third phase also cannot keep the quick descent tendency. The optimization placement is then terminated. The quick descent tendency is weighed with a local optimization number L, which is described in section 3.3.2.

3.2   The Generation of a New State

One iteration step of our algorithm is called 'the generation of a new state'. Figure 2 shows how to generate a new placement state. Only when the change in the total wire length ΔC is negative, is the new state accepted; otherwise, it is given up and a new state begins.

In step (7) in Figure 2, cell overlapping should be eliminated after a new state has been accepted, so that the accepting condition of a new placement state based on the exact total wire length change can be ensured. In our algorithm, all the effected cells are moved in rows with the overlapping length from left to right one by one. The location information is updated.

3.3   The Iterative Algorithm

A heuristic algorithm described above based on local optimization is adopted in the process of placement optimization. The main advantage of this algorithm is its fast convergence while realizing near-optimal solutions.

The operation described in section 3.2 is carried out iteratively, until there is no new placement configuration to reduce the total wire length any more, that is, a local optimum is reached. Then the next cell location adjustment cycle should be accepted unconditionally. The last placement configuration before the unconditionally accepted adjustment, perhaps, is not the best placement configuration, and the perturbed solution may yield a further

---

(1) Randomly select cell a in *row_a*
(2) Compute target location X in *row_b*
(3) **If** (a and b are in the same row)
     Compute ΔC for *intra-row* move
(4) **Else if** (adding a to *row_b* doesn't exceed length limit for *row_b*)
     Compute ΔC for moving a to location x in *row_b*
(5) **Else if** (length of neither row is exceeded)
     Compute ΔC for exchanging a and b
(6) **Else**
     Go to step (1)
(7) **If** ($\Delta C < 0$) **then**
     Change location of cell a to position X and eliminate overlaps for the rows of a and b

---

FIGURE 2   The generation of a new state.

reduction in the total wire length. Based on the heuristic placement adjustment, another local optimization goes on.

3.3.1 *The Recorder*

A recorder is set up to save and update each better local optimal placement. It also stores the best iterative optimization result when the whole placement procedure is finished. The probability of accepting the new configuration when a local optimization is reached is 100% in our algorithm; the phenomenon that the final placement result is different with each run of the placement algorithm did not exist.

3.3.2 *The Setting of a Local Optimization Number L*

In the heuristic algorithm, the total wire length is not always reduced after a local optimization and there may be some fluctuations during the whole iterative procedure as shown in Figure 3 for circuit *primary1*. Here, one black dot denotes a local optimum. Therefore, in order to offer a good trade-off between solution quality and run time, a local optimization number L is defined. Whenever the number of local optimization iteration reaches L, the current total wire length will be compared with that L times ago, and only when the former is smaller than the latter, the level selection of fixed points under use will go on, otherwise the selection will turn to the next level. The turning point is the best placement state of the performed phase, which can obtain the minimal total wire length. And if the third level selection still cannot keep the descent tendency, the optimization placement is terminated. Now, the optimal placement state in the third phase is the global optimal placement.
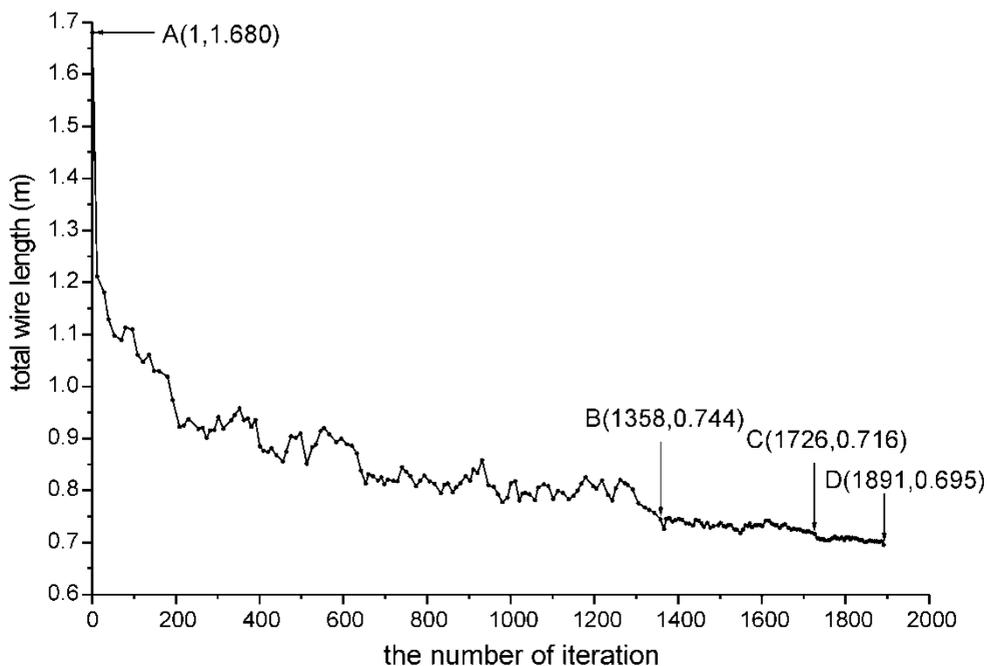

FIGURE 3   The total wire length as a function of cycle number for circuit *primary1*. A → B, B → C, C → D are the first, second, and third phases, respectively, with corresponding level selections.

How to set the value of L? If L is too small, the transition between two phases is so quick that optimization iteration in each phase will be insufficient. On the contrary, if L is too large, a great deal of time will be consumed in each phase and the optimization result will not be improved distinctly. Based on plenty of circuit experiments, we set L to be 30.

In Figure 3, point A is the beginning point of placement optimization, and points B, C and D are the optimal points in the first, second, and third phases, respectively, based on the local optimization number $L = 30$.

4   EXPERIMENTAL RESULTS

In order to compare our algorithm with others, four circuits with less than 3000 cells from the MCNC standard cell benchmarks are used. The characteristics of these benchmarks are given in Table I, where the row spacing is equal to the height of a cell. Because it is well known that the very large-scale circuits placement based on such iterative optimization algorithms as FDR or IFDR, which determines cell locations iteratively, cannot obtain good results, the IFDR algorithm needs to be combined with other approaches, such as the partitioning method for placement of large circuits. In this article, the main purpose is to verify the advantage of the proposed IFDR algorithm to perform iterative placement optimization, therefore large circuits have not been considered.

The first experiment has been done to compare our IFDR algorithm with the FDR method. Here, only the first phase with the first level selection of fixed points in the IFDR method is experimented, and the FDR method is performed with $\varepsilon = \lambda^* = 4$. The results in Table II indicate that our IFDR algorithm is more effective in finding cell target location than the FDR method.

The second experiment has been done to compare our IFDR algorithm with TW and FengShui (FS) [4]. The results given in Table III show that 16% and 17%, on average, of the total wire length can be reduced with our IFDR algorithm, compared with TW and FS, respectively. Since results for circuits *Fract* and *Struct* have not been published in

TABLE I   MCNC Standard Cell Benchmarks.

Circuit	Cells	Nets	Rows
Fract	149	163	6
Primary1	752	904	16
Struct	1888	1920	20
Primary2	2907	3029	28

TABLE II   Comparison of Wire Length (m) from FDR and IFDR.

Circuit	Primary wire length	Wire length after optimization	
		FDR	IFDR
Fract	0.073	0.034	0.025
Primary1	1.680	1.153	0.744
Struct	0.896	0.588	0.363
Primary2	10.299	5.543	3.694

TABLE III   Comparison of Wire Length (m) from TW, FS, and IFDR.

| Circuit | TW | FS | IFDR | Wire length reduction | |
				vs. TW (%)	vs. FS (%)
Fract	0.032	0.032	0.024	25	25
Primary1	0.83	1.018	0.695	17	32
Struct	0.383	0.380	0.356	7	7
Primary2	3.53	3.684	3.537	0	4

Ref. [1], we take the values from Ref. [4], which are from the commercial version of TimberWolf1.2.

5   CONCLUSION AND FUTURE WORK

In this article, a new heuristic algorithm based on the IFDR method is proposed for the iterative optimization placement problem. The developed IFDR method can provide a more efficient way for cell location adjustment and obtain better iterative optimization results than the FDR method. The combined heuristic algorithm based on local optimization can also obtain better placement results with circuits less than 3000 cells, compared with TW and FS tools.

For very large-scale circuits, there are usually some cells connected tightly with each other, while some others are connected loosely. The efficiency of finding target cell locations with the IFDR method will be greatly decreased in those loosely connected cells. Therefore, in the future work, the circuits will be first partitioned based on the clustering approach [5], and then the heuristic algorithm can be used for each sub-circuit.

References

[1] Sun, W.-J. and Sechen, C. (1995). Efficient and effective placement for very large circuits. *IEEE Trans. Computer-Aided Design*, **14**(3), 349–359.
[2] Kernighan, B. W. and Lin, S. (1970). An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, **February**, 291–307.
[3] Goto, S. (1981). An efficient algorithm for the two-dimensional placement problem in electrical circuit layout. *IEEE Trans. on Circuits and Systems*, **CAS-28**(1), 12–18.
[4] Yildiz, M. C. and Madden, P. H. (2001). Global objectives for standard cell placement. In: *Proc GLSVLSI*, pp. 68–72.
[5] Karypis, G., Aggarwal, R., Kumar, V. and Shekhar, S. (1997). Multilevel hypergraph partitioning: Application in vlsi domain. In: *Proc. Design Automation Conference*, pp. 526–529.

The Scientific World Journal

Hindawi

▸ Impact Factor **1.730**

▸ **28 Days** Fast Track Peer Review

▸ All Subject Areas of Science

▸ Submit at http://www.tswj.com