Optimal Cell Flipping in Placement and Floorplanning

Chiu-wing Sham
Department of CSE
Chinese Univ. of Hong Kong
Hong Kong
cwsham@cse.cuhk.edu.hk

Evangeline F. Y. Young
Department of CSE
Chinese Univ. of Hong Kong
Hong Kong
fyyoung@cse.cuhk.edu.hk

Chris Chu
Department of ECE
Iowa State University
USA
cnchu@iastate.edu

ABSTRACT

In a placed circuit, there are a lot of movable cells that can be flipped to further reduce the total wirelength, without affecting the original placement solution. We aim at solving this flipping problem optimally. However, solving such a problem optimally is non-trivial given the gigantic sizes of modern circuits. We are able to identify a large portion of cells (about 75%) of which the orientation (flipped or not flipped) can be determined independent of the orientations of all the other cells. We have derived three non-trivial conditions to identify those so called *independent cells*, *strictly solvable cells* and *conditionally solvable cells*. In this way, we can greatly reduce the number of cells whose orientations are dependent on each other. Finally, the cell flipping problem of the remaining *dependent cells* can be formulated as a Mixed Integer Linear Programming (MILP) problem and solved optimally. However, this may still be too slow for extremely large circuits and we have applied two other methods, Linear Programming (LP) and Linear Programming followed by Mixed Integer Linear Programming (LP+MILP) to solve the problem. Experimental results show that by identifying those independent and solvable cells first and applying the LP+MILP technique, we can solve this flipping problem effectively and obtain results just 0.01% more than the optimal. In addition, we can improve the wirelength and number of overflow tiles by 5% and 9% respectively on the floorplanning benchmarks.

Categories and Subject Descriptors: B.7.2 [Design Aids]: Placement and routing

General Terms: Algorithms, Design

Keywords: Floorplanning, Placement, Wirelength, Orientation, Flipping

1. INTRODUCTION

Many wirelength-driven placement or floorplanning frameworks have been proposed in recent years. However, the cell flipping problem is not always considered in those frameworks. In fact, some movable objects in VLSI design can be flipped or rotated. Because the flipping operations can be performed without changing the original packing, we can apply cell flipping to further reduce the total wirelength as a post-processing step while keeping the positions of all the cells unchanged. The optimal cell flipping problem has been proven to be NP-complete [3], and many heuristics were proposed to obtain sub-optimal solutions.

An analytical method [12] was proposed to obtain near optimal solutions for this cell flipping problem but Euclidean distance was used to estimate the total wirelength. A neural network approach [9] and a simulated annealing approach [10] were later proposed and competitive solutions in comparison with the analytical method could be obtained. Some simple greedy heuristics [4, 5, 6, 7] were also proposed and it could be shown that the performances of these greedy approaches were just marginally superior to those of the more complex neutral network and simulated annealing approaches. However, only two-pin nets were considered in all the above previous works and multi-pin nets were required to be broken down into two-pin nets first. A symbolic algorithm based on Boolean Decision Diagram (BDD) was proposed recently [8] that could find the optimal wirelength for small size circuits, e.g. 20-30 blocks, but it could not handle cases with large number of blocks because complete searching was performed.

In this paper, we will present a detailed study of this cell flipping problem in a packing to reduce interconnect length. We use the half-perimeter metric to measure wirelength and the latest test cases from the ISPD-05 suite [11] and MCNC benchmarks were used for the experments. The circuits are first placed by a placer or a floorplanner. We then try to identify those cells of which the orientations can be determined independent of the orientations of all the other cells. We have derived three non-trivial conditions to identify those so called *independent cells*, *strictly solvable cells* and *conditionally solvable cells*. In this way, we can greatly reduce the number of cells whose orientations are dependent on each other. Finally, the cell flipping problem of the remaining *dependent cells* cells can be formulated as a Mixed Integer Linear Programming (MILP) problem. An optimal solution can be obtained by solving it directly by a MILP solver. We will also apply two other methods: Linear Programming (LP) and Linear Programming followed by Mixed Integer Linear Programming (LP+MILP) to improve the runtime and give results that are very close to the optimal solutions.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

Notations	Descriptions
M	a set of rectangular cells
N	a set of nets
m_i	ith cell in M
n_j	jth net in N
W	total HPWL wirelength
(x_i, y_i)	center of cell m_i
$(s_{i,j}, t_{i,j})$	offset of a pin on cell m_i connecting to net n_j, with respect to the center of m_i in the x and y directions
$a_j(k)$	index of the kth cell connected by net n_j
z_j	number of cells connected by net n_j
$b_i(k)$	index of the kth net connecting cell m_i
p_i	number of nets connecting cell m_i
λ_i	horizontal orientation of cell m_i 1 : flipped, 0 : not flipped
(L_j, L'_j)	coordinates of the lower-left corner of the HPWL bounding box of net n_j
(R_j, R'_j)	coordinates of the upper-right corner of the HPWL bounding box of net n_j
LL_j	minimum possible x-coordinate of the left boundary of the HPWL bounding box of net n_j
LR_j	maximum possible x-coordinate of the left boundary of the HPWL bounding box of net n_j
RL_j	minimum possible x-coordinate of the right boundary of the HPWL bounding box of net n_j
RR_j	maximum possible x-coordinate of the right boundary of the HPWL bounding box of net n_j

Table 1: Notations

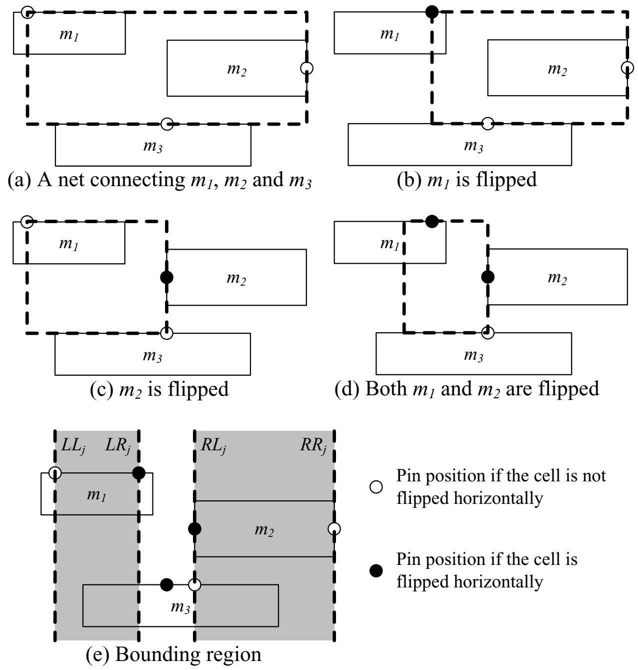

Figure 1: An Example of Bounding Box and Bounding Region of a Net n_j

2. PROBLEM FORMULATION

Given a packing of a set of rectangular cells M, we want to flip the cells horizontally or vertically without moving the centers of the cells in order to minimize the total wirelength where the total wirelength is measured by the half perimeter bounding box (HPWL) method. The goal of the cell flipping problem is to find the optimal orientations for the cells such that the total wirelength is minimized.

It is obvious that horizontal flipping does not affect the y-coordinates of the pins, and vice versa. Both horizontal and vertical flipping operations can be performed independently of each other. In the following, we will only discuss horizontal flipping, while vertical flipping can be done similarly. The notations used in following sections are summarized in table 1.

3. PRELIMINARY

An example of changing the HPWL bounding box of a net by flipping is shown in figure 1. Figure 1(a) is the initial packing and we can measure the initial HPWL. If the cell m_1 is flipped, the x-coordinate of the left boundary of the HPWL bounding box becomes larger as in figure 1(b). If the cell m_2 is flipped, the x-coordinate of the right boundary of the HPWL bounding box becomes smaller as in figure 1(c). If both m_1 and m_2 are flipped, the HPWL bounding box is changed as in figure 1(d).

We define the bounding region (LL_j, LR_j, RL_j, RR_j) of a net n_j as follows. An illustration is shown in figure 1(e). LL_j (LR_j) is the minimum (maximum) x-coordinate of the left boundary of the bounding box of net n_j. RL_j (RR_j) is the minimum (maximum) x-coordinate of the right boundary of the bounding box of net n_j. In general, the values of LL_j, LR_j, RL_j and RR_j can be evaluated as follows:

$$\begin{aligned} LL_j &= \min_{1 \le k \le z_j} \{x_{a_j(k)} - |s_{a_j(k),j}|\} \\ LR_j &= \min_{1 \le k \le z_j} \{x_{a_j(k)} + |s_{a_j(k),j}|\} \\ RL_j &= \max_{1 \le k \le z_j} \{x_{a_j(k)} - |s_{a_j(k),j}|\} \\ RR_j &= \max_{1 \le k \le z_j} \{x_{a_j(k)} + |s_{a_j(k),j}|\} \end{aligned} \quad (1)$$

4. FIXING CELL ORIENTATIONS

Today circuit designs are usually very big and involve a large number of cells. If the cell flipping problem is directly formulated and solved as a MILP, a large amount of memory and extremely long runtime will be needed. It is thus a good idea to reduce the number of cells with variable orientation first before formulating the MILP problem. We found that a lot of cells (about 75%) can actually be fixed with their optimal orientations independently of the orientations of all the other cells. According to the bounding region of each net, we can divide the cells into three types:

Independent cells: The orientations of these cells will not affect the total wirelength.

Strictly/Conditionally solvable cells: The orientations of these cells will affect the total wirelength but they can be determined optimally independent of the orientations of all the other cells.

Dependent cells: The cells that are not independent nor strictly/conditionally solvable. Their optimal orientations are dependent on the orientations of the other dependent cells.

4.1 Independent Cells

A cell m_i is independent if:

$$x_i - |s_{i,b_i(k)}| > LR_{b_i(k)} \text{ and } x_i + |s_{i,b_i(k)}| < RL_{b_i(k)} \quad (2)$$

for all $k \in \{1, 2, ..., p_i\}$. As all the possible pin positions do not overlap with the bounding regions of the corresponding nets, changes in orientations of these cells will not affect the total wirelength.

4.2 Solvable Cells

For those cells which are not dependent, their orientations will affect the total HPWL wirelength. But the orientations of some of them can still be determined optimally independent of the orientations of all the other cells. We called such kind of cells solvable cells. There are two kinds of solvable cells, strictly solvable and conditionally solvable. We will explain both of them in details in the following.

4.2.1 *Strictly Solvable Cells*

Consider a cell m_i. If it is not an independent cell, there must be at least a net n_j connecting m_i such that:

$$x_i - |s_{i,j}| \leq LR_j \text{ or } x_i + |s_{i,j}| \geq RL_j \quad (3)$$

Consider the set of nets $Q(m_i)$ connecting m_i and satisfying the above inequalities. If every net n_j in $Q(m_i)$ satisfies the condition that all the cells (except m_i) connected by n_j do not overlap with cell m_i (i.e., the x-coordinates of the pins for n_j on the other cells are all larger or smaller than that of the pin for n_j on cell m_i), m_i is called strictly solvable and the optimal orientation of m_i can be determined immediately.

This is because only the nets in $Q(m_i)$ will affect the orientation of m_i. Consider a net n_j in $Q(m_i)$. The orientation of m_i will affect the position of the left or right boundary of n_j (since m_i and n_j satisfy inequalities (3)). However, no other cells on net n_j overlap with m_i horizontally, so the effect of m_i's orientation on the wirelength of net m_j is clear, which is either $2 \times |s_{i,j}|$ or $-2 \times |s_{i,j}|$, depending whether m_i is on the left or the right boundary of n_j and the original orientation of m_i. Therefore, we can compute the total effect of m_i's orientation on W by summing up it's effects on each net in $Q(m_i)$, and determine m_i's optimal orientation.

4.2.2 *Conditionally Solvable Cells*

If a cell m_i is not strictly solvable, there must be at least one net n_j in $Q(m_i)$ such that some cells on n_j overlap with m_i horizontally. Let $Q_1(m_i)$ be a subset of $Q(m_i)$ such that every net n_k in $Q_1(m_i)$ satisfies the condition that all the cells (except m_i) connected by n_k do not overlap with cell m_i, and we denote $Q(m_i) - Q_1(m_i)$ by $Q_2(m_i)$. Following the same argument for strictly solvable cells, we can determine the effect of m_i's orientation on the total wirelength of the nets in $Q_1(m_i)$. Let this be $\delta_1(m_i)$. The value of $\delta_1(m_i)$ will determine a *potential* orientation X of m_i. Then, for each net n_j in $Q_2(m_i)$, we can determine the largest possible "adverse" effect on n_j if m_i follows the potential orientation X. This largest possible adverse effect on n_j will depend on whether m_i overlaps with the left or right bounding region of n_j and the potential orientation X. Then we can sum up all these adverse effects for each net n_j in $Q_2(m_i)$ to obtain $\delta_2(m_i)$. If $|\delta_2(m_i)| < |\delta_1(m_i)|$, cell m_i is a conditionally solvable cell and its optimal orientation should be X. Otherwise, m_i is not solvable and its orientation will be determined later.

The largest possible adverse effect $\delta_2(m_i, n_j)$ on a net $n_j \in Q_2(m_i)$ if m_i follows a potential orientation X depends on whether m_i overlaps with the left or right bounding region of n_j and the potential orientation X. There are three possible cases. In the first case, m_i overlaps with the right bounding region of net n_j only. In this case, if X is flipping m_i to the left (more exactly, it should be flipping the pin on m_i connecting to n_j to the left), $\delta_2(m_i, n_j) = 0$. Otherwise, $\delta_2(m_i, n_j)$ can be obtained by assuming that all the other cells on n_j that overlap with m_i are flipped in such a way to minimize the wirelength of n_j and $\delta_2(m_i, n_j)$ will be the difference in the wirelength of n_j between the case of flipping m_i to the right and flipping m_i to the left. The second case of m_i overlapping with the left bounding region of net n_j only can be considered similarly. In the third case, m_i overlaps with both the left and right bounding regions of net n_j. In this case, $\delta_2(m_i, n_j)$ will be the difference in the wirelength of n_j between the case when m_i follows the orientation X and all the other cells are flipped in such a way to maximize the wirelength and the case when m_i follows the opposite orientation of X and all the other cells are flipped in such a way to minimize the wirelength.

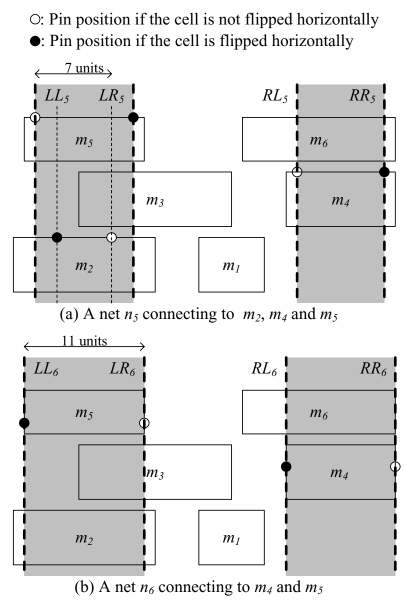

Figure 2: An Example of a Conditionally Solvable Cell m_5

An example of a conditionally solvable cell is shown in figure 2. We can look at cell m_5 in figure 2. It overlaps with the bounding regions of both net n_5 and net n_6, so it is not an independent cell. It does not overlap with the other cells on n_6 but overlaps with cell m_2 on n_5. Thus, we have $Q_1(m_5) = \{n_6\}$ and $Q_2(m_5) = \{n_5\}$. It is obvious that m_5 should not be flipped to give a smaller HPWL when considering n_6 only. Thus, the value of $\delta_1(m_5)$ for this potential orientation X of not flipping is $2 \times |s_{5,6}| = 11$. Then, we need to determine the largest possible adverse effect on $n_5 \in Q_2(m_5)$ if m_5 does not flip. The cell m_5 overlaps with the left bounding region of n_5. Since the potential orientation X will put the pin on m_5 connecting to net n_5 to the left, $\delta_2(m_5, n_5)$ will be the difference in the wirelength of n_5 between not flipping and flipping m_5 when the orientation of m_2 is such that the wirelength of n_5 is minimized, i.e., flipping the pin connecting m_2 to net n_5 to the right. This value is 7 in this example. Finally, we have $|\delta_2(m_5)| < |\delta_1(m_5)|$, so the cell m_5 is a conditionally solvable cell and its optimal orientation should be "not flipped".

After we fix the orientations of those independent cells and strictly/ conditionally solvable cells, some of the remaining cells will become independent or solvable. In our approach, we will repeatedly find independent and solvable cells to fix their orientations until there are less than ten new independent or solvable cells found in one iteration. For those remaining cells of which the orientations cannot be fixed yet (called dependent cells), we will formulate a MILP to determine their orientations. The numbers of cells of which the orientations can be fixed after this cell orientation fixing step are shown in table 2. With this orientation fixing step, about $70 - 80\%$ of cells can get their optimal orientations fixed immediately.

Test cases	Orientation fixing step		
	No. of cells	No. of fixed cells	Percentage
*adaptec*2	255023	184009	72.15%
*adaptec*4	496045	374140	75.42%
*bigblue*1	278164	184207	66.22%
*bigblue*2	557866	408528	73.23%
*bigblue*3	1096812	873452	79.64%
*bigblue*4	2177353	1665630	76.50%
Average	$---$	$---$	73.86%

Table 2: Percentage of Cells Got Fixed with Optimal Orientations after the Orientation Fixing Step

5. SOLVING DEPENDENT CELLS

In this section, we will describe how we formulate the cell flipping problem for the remaining dependent cells as a MILP. A pseudo code of the whole process is shown in figure 3. Now, we focus on the step of finding the optimal orientations for the remaining dependent cells (step 3). Consider a net n_j connecting a set of z_j cells, the coordinates of the pin on the cell m_i connecting to n_j where $i \in \{a_j(1), a_j(2), ..., a_j(z_j)\}$ can be represented by:

$$(x_i + s_{i,j}(1 - 2\lambda_i), y_i + t_{i,j}(1 - 2\lambda'_i))$$

The variable λ_i tells whether cell m_i will be flipped horizontally and λ'_i tells whether cell m_i will be flipped vertically. If m_i is flipped horizontally (vertically), λ_i (λ'_i) is 1; otherwise, it is 0. Because all the pins connecting to n_j should be located inside the bounding box of n_j, we have:

$$\begin{aligned} x_i + s_{i,j}(1 - 2\lambda_i) &\leq R_j \\ x_i + s_{i,j}(1 - 2\lambda_i) &\geq L_j \\ y_i + t_{i,j}(1 - 2\lambda'_i) &\leq R'_j \\ y_i + t_{i,j}(1 - 2\lambda'_i) &\geq L'_j \end{aligned} \tag{4}$$

for all $i \in \{a_j(1), a_j(2), ..., a_j(z_j)\}$. Finally, we want to minimize the total wirelength of the packing, so we have the objective function as follows:

$$W = \sum_{j=1}^{|N|} (R_j - L_j + R'_j - L'_j) \tag{5}$$

It is easy to see that it is a MILP with a set of linear inequality constraints (4) and a linear objective function (5). We can solve it optimally by invoking a MILP solver. However the runtime for extremely large test cases will still be un-affordable even after fixing the orientations of about 75% of cells.

We have tried three different methods to solve the above MILP problem. We can obtain optimal solution by solving it directly with a MILP solver. We can also solve it more efficiently as a LP to obtain a near optimal solution. The best compromise will be solving it with a combination of LP and MILP. We will discuss each of them with details in the following sections. We have also implemented a simple greedy approach for comparison purpose.

1 Placement results from placer
2 Cell orientation fixing step
    Repeat
        Orientation fixing for ***independent*** cells
        Orientation fixing for ***strictly/conditionally solvable*** cells
    Until (no. of newly fixed cells < 10)
3 Find the orientations of remaining ***dependent*** cells
    Formulate the cell flipping problem as **MILP**
    Solve the problem by **MILP**, **LP**, or **LP+MILP**

Figure 3: The Design Flow of our Approach

5.1 MILP

The MILP can be solved separately in the horizontal and vertical directions. We assume the x-direction in the following discussion and the y-direction can be handled similarly. The MILP in the x-direction is formulated as follows:

Minimize:

$$W = \sum_{j=1}^{|N|} (R_j - L_j) \tag{6}$$

Subject to

$$\begin{aligned} x_i + s_{i,j}(1 - 2\lambda_i) &\leq R_j \\ x_i + s_{i,j}(1 - 2\lambda_i) &\geq L_j \end{aligned} \tag{7}$$

$\forall j \in \{1, 2, ..., |N|\}$ and $\forall i \in \{a_j(1), a_j(2), ..., a_j(z_j)\}$. Although we can obtain the optimal solution, the runtime is slow and the memory usage is huge. However, the optimal solution can be used as a reference point to see how good the other approaches are.

5.2 LP

With the same formulation, we can solve the problem by LP instead. When there are real numbers in the solution, we can simply round the solution to 1 or 0:

$$\begin{aligned} \lambda_i &= 1 \text{ if } \lambda_i > 0.5 \\ \lambda_i &= 0 \text{ otherwise} \end{aligned} \tag{8}$$

From the experimental results, we found that more than half of the λ_is will be set to either one or zero after solving the LP. The result is quite close to the optimal but the runtime will be much shorter.

5.3 LP+MILP

In order to have a good balance between runtime and solution quality, we can combine the LP and MILP approaches. As mentioned above, we found from the experimental results that a significant portion of λ_is will be set to either zero or one after solving the LP. We can then fix the orientations of those cells and formulate another MILP for the remaining cells. This combined approach will be slower than the LP approach but solutions very close to the optimal can be obtained.

5.4 Greedy Approach

In the greedy approach, we will try to flip each cell one after another to obtain a smaller total wirelength. In each iteration, every cell is considered once. A cell is flipped if and only if flipping that cell will lead to a smaller total wirelength. This process is repeated as long as the total improvement on W in one iteration is more than 0.02%. Notice that in each step of computing the orientation of a cell, only the wirelengths of the nets which are connected to the cell under consideration are required to be calculated.

6. EXPERIMENTAL RESULTS

In the first set of experiments, the test cases used are the ISPD-05 suite circuits [11]. Detailed information of the testing circuits are shown in table 3. The pre-placed circuits are obtained from the ISPD-05 placement contest web site. We will apply our approaches on all the placement results and show the average runtime and the improvement in wirelength.

Test cases	No. of cells	No. of nets	No. of pins
*adaptec*2	255023	254457	1069482
*adaptec*4	496045	494716	1912420
*bigblue*1	278164	277604	1144691
*bigblue*2	557866	534782	2122282
*bigblue*3	1096812	1095519	3833218
*bigblue*4	2177353	2169183	8900078

Table 3: Information of the Testing Circuits

We used CPLEX9.0 as the LP and MILP solver. The tolerance was set to 0.02%. It means that the solutions obtained from the solver are at least 0.02% close to the optimal solutions. All programs were written in C and were run on a Sun Blade 2500 workstation with a 1.6GHz UltraSPARC IIIi CPU and 2GB RAM.

A summary of the improvement is shown in table 4. Because the circuits are pre-placed by wirelength-driven placers, the optimal improvement is around $1-2\%$ on average. We can see that the solutions by LP+MILP are very close to the optimal solutions. The improvement by LP is about 10% worse than the optimal improvement. The improvement by the greedy approach is about 26% worse than the optimal improvement. Note that we cannot obtain a result by MILP nor LP+MILP on the circuit *bigblue*4 because of the excessive memory usage. Table 5 shows the runtime of different methods. We can see that the runtime is improved significantly when LP is applied.

In the second set of experiments, the test cases used are the MCNC benchmark circuits. Detailed information of the testing circuits is shown in table 6. The pre-placed circuits are obtained from the floorplanner Parque [2]. We will apply our approaches on all the floorplan results and show the average runtime and the improvement in wirelength and routability by global routing [1].

A summary of the improvement in HPWL is shown in table 7. The optimal improvement is around $3\% - 10\%$ on average. We can see that the solutions by LP+MILP are very close to the optimal solutions. The improvement by LP is on average 4% worse than the optimal improvement. The improvement by the greedy approach is on average 20% worse than the optimal improvement. In addition, We can also see that the difference between the optimal HPWL and the HPWL obtained from the floorplanner is quite large.

Test cases	With orientation fixing (s)			Greedy (s)
	MILP	LP+MILP	LP	
*adaptec*2	1688.45	717.66	158.95	37.29
*adaptec*4	2996.40	1000.15	330.57	79.33
*bigblue*1	1707.92	417.06	160.32	37.91
*bigblue*2	6567.07	2313.76	554.96	409.02
*bigblue*3	11522.79	2415.38	809.83	240.46
*bigblue*4	$---$	$---$	3854.34	1478.53
w.r.t. LP	11.15	3.48	1.00	0.36

Table 5: Runtime of Different Methods

Test cases	No. of cells	No. of nets	No. of pins	No. of pads
hp	9	97	264	73
xerox	10	203	696	2
apte	11	83	214	45
*ami*33	33	123	480	42
*ami*49	49	408	931	22

Table 6: Information of the Testing Circuits

To study further the effect of flippings on wirelength and routability, the floorplans before and after the optimal flipping step will be routed by a global router and the results on wirelength and routability are shown in table 8 and table 9 respectively. We can see that there is about 5% improvement on total wirelength and about 10% reduction in the number of overflow tiles in global routing with optimal flipping. This shows that we can also improve the routability significantly by optimal block flipping.

Test cases	Wirelength from global routing (nm)	
	No optimal flipping	W/ optimal flipping
hp	131343.60	126122.60
xerox	704591.80	668223.70
apte	339564.30	318032.50
*ami*33	60073.60	54135.60
*ami*49	1237478.00	1171123.40
Average	494610.26 (1.00)	467527.56 (0.95)

Table 8: Wirelength after Global Routing

Table 10 shows the runtime of different methods. We also compare with the results of the BDD based approach [8]. Notice that both our method and the BDD based approach can obtain optimal HPWL with flipping. However, the runtime of our method is much faster than that of the BDD based approach. It is because the orientation fixing step can reduce the problem size significantly and solving the remaining mixed integer linear programming problem becomes much easier. The BDD based approach cannot be applied to those large placement benchmark circuits.

Test cases	Reduction in total wirelength											
	MILP			LP+MILP			LP			Greedy		
	x-dir.	y-dir.	total	x-dir.	y-dir.	total	x-dir.	y-dir.	total	x-dir.	y-dir.	total
*adaptec*2	0.81%	0.43%	1.24%	0.80%	0.43%	1.23%	0.75%	0.39%	1.14%	0.65%	0.37%	1.02%
*adaptec*4	0.97%	0.55%	1.52%	0.96%	0.55%	1.51%	0.92%	0.52%	1.44%	0.80%	0.45%	1.25%
*bigblue*1	0.79%	0.68%	1.47%	0.79%	0.67%	1.46%	0.78%	0.61%	1.39%	0.72%	0.56%	1.28%
*bigblue*2	1.03%	0.77%	1.80%	1.03%	0.76%	1.79%	0.98%	0.73%	1.71%	0.84%	0.60%	1.44%
*bigblue*3	0.87%	0.46%	1.33%	0.87%	0.46%	1.33%	0.80%	0.44%	1.24%	0.58%	0.34%	0.92%
*bigblue*4	− − −	− − −	− − −	− − −	− − −	− − −	0.75%	0.34%	1.09%	0.55%	0.25%	0.80%
Average	0.89%	0.58%	1.47%	0.89%	0.58%	1.47%	0.83%	0.51%	1.34%	0.69%	0.43%	1.12%
w.r.t MILP	1.00	1.00	1.00	0.99	1.00	1.00	0.92	0.86	0.90	0.75	0.71	0.74

Table 4: A Summary of the Improvement on Total Wirelength by Different Methods on the Placement Benchmarks

Test cases	Reduction in total wirelength											
	MILP			LP+MILP			LP			Greedy		
	x-dir.	y-dir.	total	x-dir.	y-dir.	total	x-dir.	y-dir.	total	x-dir.	y-dir.	total
hp	1.75%	1.90%	3.65%	1.75%	1.90%	3.65%	1.69%	1.85%	3.54%	1.55%	1.57%	3.13%
xerox	2.29%	2.75%	5.04%	2.29%	2.74%	5.03%	2.28%	2.74%	5.02%	1.93%	2.56%	4.48%
apte	2.66%	2.48%	5.14%	2.64%	2.48%	5.11%	2.64%	2.48%	5.11%	2.27%	2.18%	4.45%
*ami*33	5.01%	5.25%	10.26%	5.01%	5.25%	10.26%	4.74%	4.64%	9.38%	3.88%	3.65%	7.53%
*ami*49	2.93%	2.49%	5.42%	2.91%	2.44%	5.35%	2.91%	2.40%	5.30%	2.30%	1.89%	4.19%
Average	2.93%	2.97%	5.90%	2.92%	2.96%	5.88%	2.85%	2.82%	5.67%	2.39%	2.37%	4.76%
w.r.t MILP	1.00	1.00	1.00	0.99	1.00	1.00	0.97	0.95	0.96	0.81	0.79	0.80

Table 7: A Summary of the Improvement on Total Wirelength by Different Methods on the Floorplanning Benchmarks

Test cases	No. of overflow tiles from global routing	
	No optimal flipping	W/ optimal flipping
hp	13.00	13.00
xerox	27.00	25.00
apte	17.00	16.00
*ami*33	11.00	10.00
*ami*49	43.00	37.00
Average	22.20 (1.00)	20.20 (0.91)

Table 9: No. of Overflow Tiles after Global Routing

Test cases	With orientation fixing (s)			Greedy (s)	BDD based[1] (s)
	MILP	LP+MILP	LP		
hp	0.01	0.01	0.01	0.01	0.90
xerox	0.14	0.13	0.06	0.01	2.10
apte	0.06	0.05	0.03	0.01	0.90
*ami*33	0.06	0.07	0.04	0.01	2.30
*ami*49	0.22	0.20	0.08	0.01	19.00
w.r.t. LP	2.23	2.09	1.00	0.22	113.66

[1]Using Intel P4 with 3GHz CPU and 2G memory.

Table 10: Runtime of Different Methods

7. CONCLUSION

We have presented a detailed study on the optimal cell flipping problem in placement and floorplanning. Experiment results show that we can obtain solutions very close to the optimal by the LP+MILP approach together with the orientation fixing step. Results show that the runtime of this cell flipping step is short in comparison with the runtime of the placement or floorplanning step. Our approach is thus desirable to be applied as a post-processing step in placement or floorplanning to improve wirelength and routability.

8. REFERENCES

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

[2] S. N. Adya and I. L. Markov. Consistent placement of macro-blocks using floorplanning and standard-cell placement. In *Proceedings of Internation Symposium on Physical Design*, pages 12–17, 2002.

[3] C. K. Cheng, S. Z. Yao, and T. C. Hu. The orientation of modules based on graph decomposition. In *IEEE Transactions on Computers*, pages 774–780, June 1991.

[4] K. Chong and S. Sahni. Flipping modules to minimize maximum wire length. In *Proceedings of International Conference on Computer Design: VLSI in Computers and Processors*, 1991.

[5] K. Chong and S. Sahni. Minimizing total wire length by flipping modules. In *Proceedings of International Conference on VLSI Design*, pages 25–30, 1992.

[6] K. Chong and S. Sahni. Minimizing total wire length by flipping modules. In *IEEE Transactions on CAD of Integrated Circuits and Systems*, pages 167–175, January 1993.

[7] K. Chong and S. Sahni. Flipping modules to improve circuit performance and routability. In *Proceedings of International Conference on VLSI Design*, pages 127–132, 1994.

[8] X. Hao and F. Brewer. Wirelength optimization by optimal block orientation. In *Proceedings of IEEE Internation Conference on Computer-Aided Design*, 2005.

[9] R. Libeskind-Hadas and C. L. Liu. Solutions to the module orientation and rotation problems by neutral computation network. In *Proceedings of the 26th ACM/IEEE Design Automation Conference*, pages 400–405, 1989.

[10] S. Nahar, E. Shragowitz, and S. Sahni. Simulated annealing and combinatorial optimization. In *International Journal of Computer Aided VLSI Design*, pages 1–23, 1989.

[11] G. J. Nam, C. J. Alpert, P. Villarubbia, B. Winter, and M. Yildiz. The ispd2005 placement contest and benchmark suite. In *Proceedings of Internation Symposium on Physical Design*, pages 216–220, 2005.

[12] M. Yamada and C. L. Liu. An analytical method for optimal module orientation. In *Proceedings of IEEE Internation Symposium on Circuits and Systems*, pages 1679–1682, 1988.