

**Computer Science Department Technical Report
University of California
Los Angeles, CA 90024-1596**

**A PROVABLE NEAR-OPTIMAL ALGORITHM FOR THE
CHANNEL PIN ASSIGNMENT PROBLEM**

**Jason Cong
Kei-Yong Khoo**

**March 1991
CSD-910008**

A Provable Near-Optimal Algorithm for the Channel Pin Assignment Problem

Jason Cong and Kei-Yong Khoo

*Department of Computer Science
University of California at Los Angeles
Los Angeles, CA 90024, U. S. A.*

Abstract

In this paper, we present a near-optimal algorithm for the channel pin assignment problem. Our algorithm, called the alternative packing algorithm, can always produce an assignment solution whose density is at most one more than the density of an optimal solution (i.e., $d(S) \leq d(S^*) + 1$, where $d(S)$ is the density of our solution and $d(S^*)$ is the density of the optimal solution). We tested our algorithm on a number of channel routing benchmark examples and achieved significant reduction in channel density and total net span. Our algorithm has important application to the pin assignment and global routing problems in general cell layout design since it has been shown that the combined pin assignment and global routing problem can be reduced to a series of channel pin assignment problems.

1. Background and Motivation

The channel routing problem is an important problem in VLSI layout design. The conventional channel routing problem have been studied extensively, where the positions of the pins on the bottom and top edge are fixed (for example, see [De76, YoKu82, FiRi82, BuPe83]). However, it is possible in practice to have channel routing problems in which pin positions are not fixed. For example, if the pins on one or both edges of a channel are output pins of a PLA, then these pins are permutable. In this case, we want to determine the position of each pin in the channel so that the resulting channel density is minimum. This problem is called the channel pin assignment problem.

One strong motivation of studying the channel pin assignment problem is for solving the pin assignment and global routing problems in general cell layout design since it has been shown that the combined pin assignment and global routing problem can be eventually reduced to a series of channel pin assignment problems based on the results in [Co89, PeMK90, Co91]. The results in [Co89, Co91] can be described briefly as follows: In order to obtain a high-quality pin assignment solution in general cell layout design, pin assignment and global routing are carried out simultaneously. Since the complexity of the combined problem could be very high, the block boundary decomposition technique was introduced, which divides the boundary of each block into a set of intervals such that each interval is adjacent to exactly one channel. Fig. 1 shows the block boundary decomposition of a simple floorplan, which can be computed easily from the corresponding channel intersection graph [Pr79]. Under such block boundary decomposition, a coarse pin assignment solution is defined to be a mapping of each pin to an interval on the

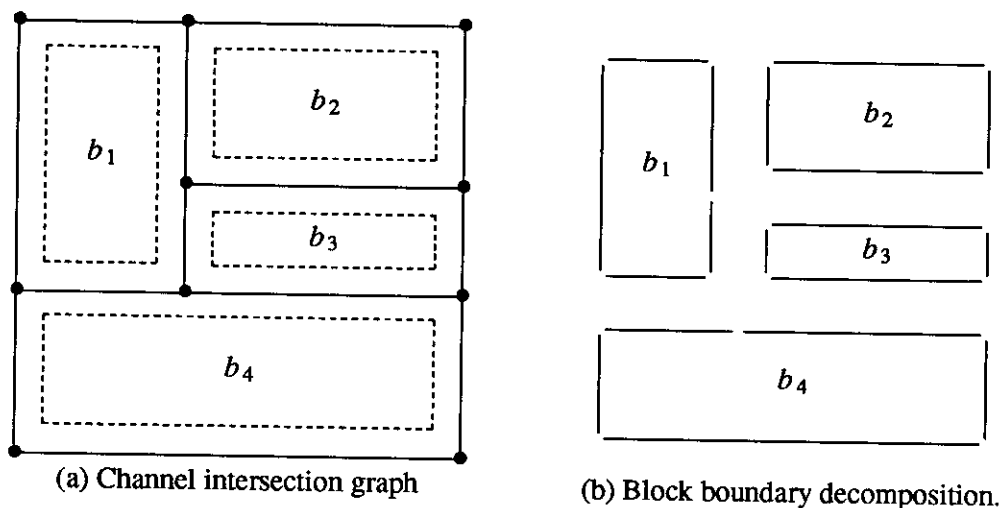


Figure 1: Block boundary decomposition of a given floorplan.

boundary of its corresponding block (note that the exact pin positions are not given in a coarse pin assignment solution). After the block boundary decomposition, the algorithm computes a combined pin assignment and global routing solution in two steps. The first step is to compute a coarse pin assignment solution together with a global routing solution (i.e., we determine the interval assigned for each pin and the channels for connecting each net). Then, the second step is to compute a refinement of the coarse pin assignment and global routing solution obtained in the first step so that the exact position of each pin in its interval. Fig. 2 (a) shows a coarse pin assignment and global routing solution for a net computed in the first step (where $g(p_i)$ specifies the interval that pin p_i is assigned to), and Fig. 2 (b) shows a refinement of the coarse pin assignment and global routing solution after in the second step. It was shown that an optimal refinement (in terms of minimizing total chip area) can always be achieved in the second step provided that we can solve the channel pin assignment problem optimally. Therefore, we are very interested in the optimal solution to the channel pin assignment problem.

An optimal channel pin assignment algorithm was presented in [Co89, Co91] for basic channels in which each net has at most two pins. If the given channel has no exit, it was shown in [LeLi85] that the an optimal channel pin assignment can be computed in linear time. A generalized channel pin assignment problem was studied in [CaWo90] in which both position and order constraints are considered. The resulting problem is shown to be NP-complete and a polynomial time solution was presented when the relative orderings of the pins are completely fixed. The maximum pin alignment problem, the via minimization problem and the river routing detection problem were studied for the channels with movable pins in [WiWo85, DeBh89, ToTr90] assuming that there are no exits in the channel. However, the problem of computing an optimal channel pin assignment for general channels with exits was open. In this paper, we present an $O(m \log m)$ time near-optimal channel pin assignment algorithm for general channels with exits. Our algorithm, called the alternative packing algorithm, can always produce a solution whose density is no more than the density of an optimal solution plus one (i.e., $d(S) \leq d(S^*) + 1$, where $d(S)$ is the density of our solution and $d(S^*)$ is the density of the optimal solution).

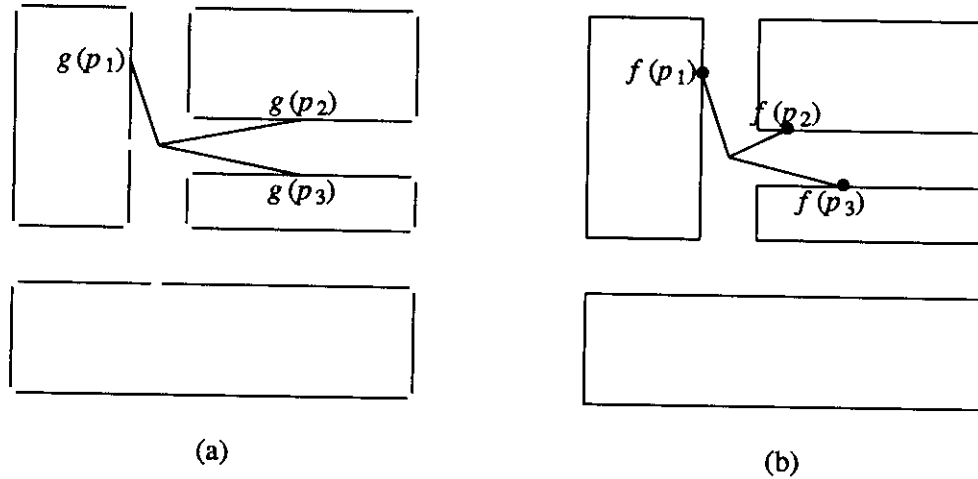


Figure 2: (a) A coarse pin assignment and global routing solution of a net;
 (b) A refinement of the coarse pin assignment and global routing solution.

Clearly, this result can be applied to compute a near-optimal refinement of a coarse pin assignment and global routing solution according to the above discussion. Moreover, our algorithm can also be utilized in the approach presented in [PeMK90] where floorplanning, pin assignment, and global routing are all computed simultaneously based on a hierarchical structure.

The remainder of this paper is organized as follows: In Section 2, we formulate the problem. In Section 3, we describe the algorithm in detail and outline analysis of the performance bound. Experimental results are presented in Section 4.

2. Formulation of the Problem

Given the lists of pins on the bottom and top edges of a channel and the lists of exits of nets at the left and right sides of the channel, the *channel pin assignment problem* is to determine the pin positions on the bottom and top edges of the channel so that the resulting channel density is minimum. For the convenience of discussion, we assume that there is a grid structure imposed on the channel, although our algorithm also applies to gridless channels. Our algorithm is independent of the number of routing layers in the channel.

We shall remove all the nets with both left and right exits from the channel since the assignment of the pins in these nets can be arbitrary (each of these nets always increase the resulting channel density by one regardless how we assign the pins in that net). We shall assign the pins in these nets arbitrarily to the remaining positions after we compute an optimal (or near-optimal) assignment of the other of nets (i.e., nets with at most one exit). Also, without loss of generality, we assume that the number of bottom pins equals the number of top pins; otherwise, we can introduce a number of dummy nets, each with a single pin, to make up the difference.

3. Description of the Algorithm

It is not difficult to show that several simple heuristics for the channel pin assignment problem can produce solutions that are far from optimal. For example, one commonly thought

heuristic is to assign the pins of the same net adjacent to each other. However, Fig. 3 shows an example where the density of the solution obtained using such a heuristic is much higher than the density of an optimal solution. Our algorithm generalizes the channel pin assignment algorithm for channels without exits in [LeLi85]. The presence of exits in the channel significantly complicates the problem since the positions of exits, unlike ordinary pins which can be assigned anywhere, are fixed. In order to obtain near optimal solutions, our algorithm will first preprocess all the nets and then partition them into several classes.

For a net n , let $pt(n)$ and $pb(n)$ be the number of pins to be assigned along the top and bottom edge of the channel respectively. Let $surplus(n) = |pt(n) - pb(n)|$. We say that net n is a *top surplus net* if $pt(n) \geq pb(n)$ or a *bottom surplus net* if $pt(n) \leq pb(n)$. Notice that if $pt(n)$ equals $pb(n)$, net n can be treated as either a top surplus net or a bottom surplus net. Moreover, we call a net with a left exit to be a *left net*. We call a net with a right exit to be a *right net*. A net with no exit is called a *neutral net*. Therefore, all the nets can be classified into 6 classes as shown in Table 1.

	Left exit	No exit	Right exit
$pt(n) \geq pb(n)$	top left net	top neutral net	top right net
$pt(n) \leq pb(n)$	bottom left net	bottom neutral net	bottom right net

Table 1: Classification of nets

Before any assignment, our algorithm sorts all the nets into a list L according to their net classes (details of the ordering will be described later). Our algorithm assign pins in the channel from left to right processing one net at a time. During the execution of the algorithm, *the surplus in the channel*, denoted $surplus(channel)$, is defined to be the difference of the number of top pins and the number of bottom pins currently in the channel. We say that there is *top surplus in the channel* if there are more top pins currently in the channel and we say that there is *bottom surplus in the channel* if there are more bottom pins currently in the channel. At any step of the assignment, the positions of the surplus pins in the channel are not yet fixed while the positions of the pins left to the surplus pins are completely determined and will not be changed in the subsequent steps. There are three cases for choosing the next net for assignment. If there is no surplus in the channel, our algorithm removes the first net from L for the assignment in the next

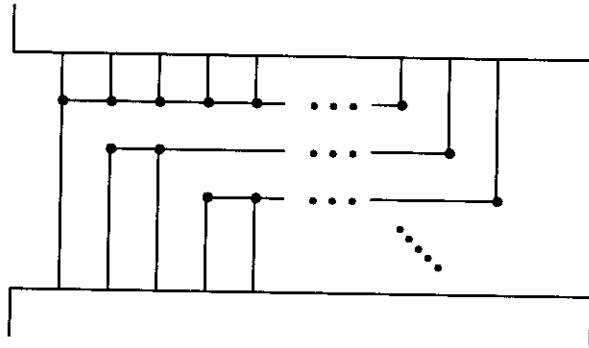


Figure 3: Assigning the pins of the same net adjacent to each other can result in large channel density

step; if there is top surplus in the channel, our algorithm will remove the first a bottom surplus net from L for assignment in the next step; if there is bottom surplus in the channel, our algorithm will remove the first a top surplus net from L for assignment in the next step. Note that the surplus type of the next net is always opposite to the surplus type in the channel. That is why our algorithm is called the *alternative packing algorithm*. Intuitively, our algorithm tries to use the surplus of next net to cancel the current surplus in the channel so that we can always keep the surplus in the channel small. Such an alternative packing scheme guarantees that at any time the surplus pins in the channel belong to the same net (we shall prove this later in the proof of Lemma 1).

Now we describe the assignment of a net in each step. Assume that there is top surplus in the channel (the case when there is bottom surplus in the channel is symmetric). Suppose that the top surplus pins belong to net n' (see Fig. 4(a)). Our algorithm will choose a bottom surplus net, say net n , for assignment in the next step. There are two cases. (i) $surplus(channel) \leq surplus(n)$. In this case, we assign all the $surplus(channel)$ top surplus pins of net n' to the leftmost available positions on the top edge. Next, we assign $surplus(channel)$ bottom pins of net n to the leftmost available positions on the bottom edge. Then, we assign each pair of matched bottom and top pins of net n to the leftmost columns in the channel. At the end, there are $surplus(n) - surplus(channel)$ bottom pins of net n left not assigned and the channel becomes bottom surplus (see Fig. 4(b)); (ii) $surplus(channel) > surplus(n)$. In this case, we assign $surplus(n)$ top surplus pins of net n' to the leftmost available positions on the top edge. Next, we assign $surplus(n)$ bottom pins of net n to the leftmost available positions on the bottom edge. Then, we assign each pair of matched bottom and top pins of net n to the leftmost columns in the channel. At the end, there are $surplus(channel) - surplus(n)$ top surplus pins of net n' left not assigned and the channel still has top surplus (see Fig. 4(c)). In both cases, the surplus pins in the channel are of the same net (provided the surplus pins in the preceding step are of the same net).

The nets in L are sorted as follows at the beginning of the algorithm. Since we assign pins from left to right in the channel, we place all the left nets before any neutral net and all the neutral nets before any right net in the list. This ensures that a net which has an exit is assigned as close to the side of the exit as possible. Moreover, we sort the left nets in the list by their increasing surplus values and the right nets in the list by their decreasing surplus values. Such an ordering minimizes the overlap of the nets with left exits and the nets with right exits. Fig. 5 shows the effect of sorting left nets and right nets by their surplus values. (This effect is very important to the proof of Lemma 2.) The algorithm is summarized in Fig. 6 below.

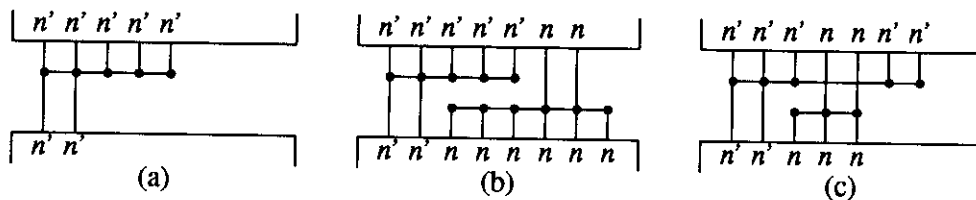


Figure 4: Assigning a bottom surplus net where the channel has top surplus.

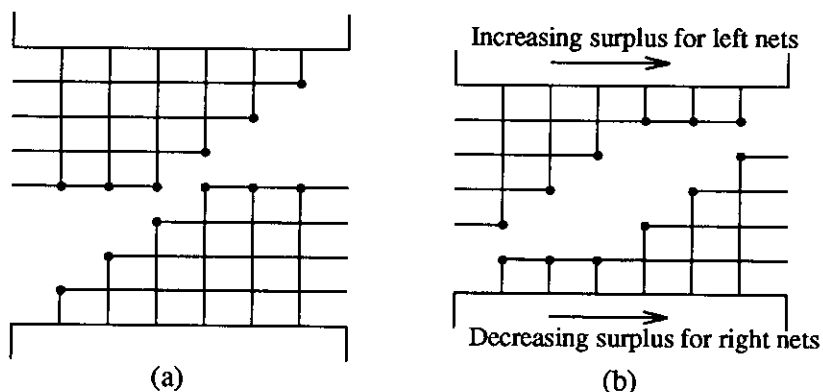


Figure 5: (a) Pins assignment made without sorting nets by their surplus values.
 (b) Pins assignment made with nets sorted properly by their surplus values.

Given a channel pin assignment solution, the *pin interval* of a net in the channel is the interval that spans from the leftmost pin to the rightmost pin of that net in the channel. The *exit interval* of a net, if the net has an exit, is the interval that spans from the left side of the channel to the leftmost pin of the net (if the net has a left exit) or from the rightmost pin of the net to the right side of the channel (if the net has a right exit). The performance bound of our alternative packing algorithm can be concluded from the following results.

Lemma 1 Given a channel pin assignment problem, the density of all the pin intervals in the solutions produced by the alternative packing algorithm is at most 2.

Proof: First, we show by induction that at any step of our algorithm, the surplus pins in the channel (the pins whose position are not yet fixed) are of the same net. Clearly, when we add the first net to the channel, the surplus pins are of the same net. Suppose that at some step, all the surplus pins in the channel belong to net n' . According to our algorithm, after we assign the next net n , the resulting surplus pins in the channel are either all of net n or all of net n' . (see Fig. 4.). Therefore, by induction, at any step of our algorithm, the surplus pins in the channel are of the same net. Since we pack only one net with the surplus pins, the density of the pin intervals in the channel is at most 2 at any step -- one due the net containing the surplus pins in the channel and another due the current net being assigned. ■

Lemma 2 Given a channel pin assignment problem with exits, the density of all the exit intervals in the solution produced by the alternative packing algorithm is at most $\max(e_l, e_r) - 1$, where e_l and e_r are the number of left and right exits, respectively.

Proof: We first introduce the notion of a concave staircase. A staircase is called a *concave staircase* if when we join the two ends of the staircase by a straight line, the entire staircase lies on one side of the straight line (see Fig. 7(a)). We can 'fold' two staircases of the same width by rotating one staircase by 180 degree and placing it on top of the other staircase so that the baselines of the two staircases are aligned (see Fig. 7(b) and 7(c)). The concave staircases have the property that if we 'fold' the two concave staircases, the height of the resulting figure is the larger of the heights of the two staircases (see Fig. 7(b)). This property is not necessarily true for arbitrary staircases since if we 'fold' two non-concave staircases, the height of resulting figure

Algorithm *Alternative_Packing*.

L_1 = all the left nets sorted by their increasing surplus values;

L_2 = all the neutral nets;

L_3 = all the right nets sorted by their decreasing surplus values;

$L = L_1$ concatenate L_2 concatenate L_3 ;

while L is not empty

if channel has top surplus

 let n = first bottom surplus net from L

else if channel has bottom surplus

 let n = first top surplus net from L

else /* if there is no surplus in the channel */

 let n = first net in L ;

endif

 Let n' = net of the surplus pins in the channel;

if $surplus(channel) < surplus(n)$

 assign all the surplus pins of net n' to the leftmost available positions;

 assign $surplus(channel)$ pins of net n to the leftmost available positions;

 assign each pair of matched bottom and top pins of net n to the leftmost available column;

 /* We now have surplus pins of net n in the channel. */

else

 assign $surplus(n)$ surplus pins of net n' to the leftmost available positions;

 assign $surplus(n)$ pins of the net n to the leftmost available positions;

 assign each pair of matched bottom and top pins of net n ;

 /* We still have surplus pins of net n' in the channel. */

endif

 remove net n from L ;

end.

Figure 6: The alternate packing algorithm

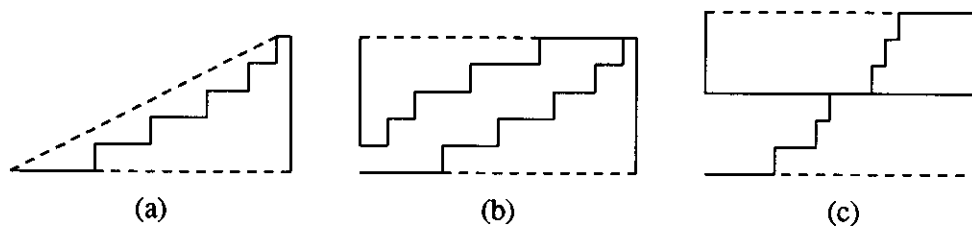


Figure 7: (a) A concave staircase. (b) 'Folding' of two concave staircases.
(c) 'Folding' of two non-concave staircases.

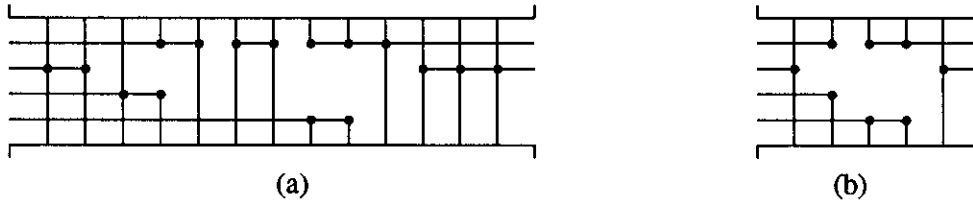


Figure 8: (a) Original channel. (b) Channel with columns removed without changing the exit intervals density.

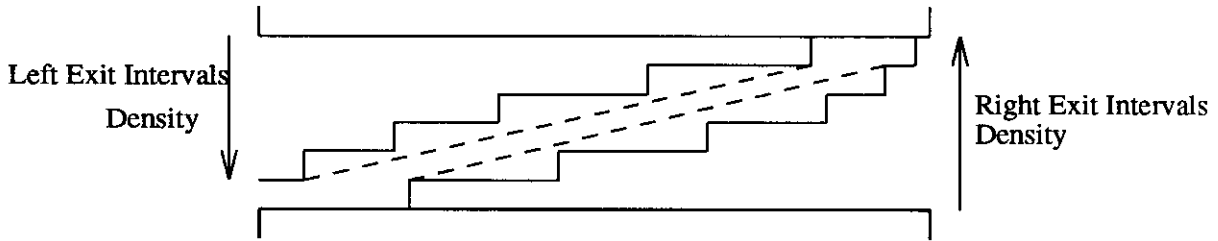


Figure 9: Density of exit intervals in the channel

could be as high as the *sum* of the heights of the two staircases as shown in Fig. 7(c).

For a channel pin assignment solution produced by the alternative packing algorithm, we can remove some columns as follows without changing the density of exit intervals: (i) For each *non-surplus* net n_i which has an exit, we remove all but one column in which both the top and bottom pins are of net n_i ; (ii) For each of the remaining net n_j , we remove all the columns in which both the top and bottom pins are of net n_j . The result is a shortened channel with the same exit interval density (as well as the same channel density). Fig. 8 shows how to obtain the shortened channel from a given channel example. In the shortened channel, except for the non-surplus nets with exits which occupies exacty one column each, the channel now contains only the surplus of each nets.

The shorten channel has the nice property that when we draw the density distributions for the left exit intervals and the right exit intervals separately, they form two folded concave staircases as shown in Fig. 9. The density distribution of left (right) exit intervals form a concave staircase because all the left (right) exit nets are sorted according to their decreasing (increasing) surplus value. For the right exit intervals, the density distribution starts from zero from the left edge of the channel and increases to $e_r - 1$ at the right edge of the channel (since the exit interval of the rightmost right exit net is null). Similarly, for the left exit intervals, the density distribution begins with $e_l - 1$ at the left edge of the channel (since the exit interval of the leftmost left exit net is null) and decreases to zero at the right edge of the channel. Clearly, the density distribution of all the exit intervals corresponds to the 'folding' of the two staircases of the left exit interval density distribution and the right exit interval density distribution. Since both staircases are concave staricases, the height of the folded result is the larger of the heights of the two staircases. Therefore, the maximum exit interval density is $\max(e_l - 1, e_r - 1) = \max(e_l, e_r) - 1$. ■

According to these two lemmas, we can conclude the following result:

Theorem Given a channel pin assignment problem with m pins, the alternative packing algorithm produces a channel pin assignment solution in $O(m \log m)$ time so that the density of the solution is at most one more than the density of the optimal solution.

Proof: Let d^* be the channel density of the optimal solution and d be the channel density of our solution. For the case where there are no exits, d^* is at least 1. (We ignored the trivial case where $pt(n) = pb(n)$ for every net n in the channel which results in d^* equals 0). According to Lemma 1, d is at most 2. Therefore $d \leq d^* + 1$. For the case where there are exits in the channel, d^* is clearly lower-bounded by $\max(e_l, e_r)$. Since the channel density is the sum of the pin interval density and the exit interval density, according to Lemma 1 and Lemma 2, $d \leq 2 + \max(e_l, e_r) - 1 = \max(e_l, e_r) + 1$. Thus, in this case we still have $d \leq d^* + 1$. Therefore in general, $d \leq d^* + 1$.

The algorithm first sorts the nets according their surplus value, which takes $O(m \log m)$. Then, the algorithm assigns the nets one by one. The assignment of net n_i takes $O(m_i)$ time where m_i is the number of pins in net n_i . Therefore, the assignment of all the nets takes $O(m)$ time. Thus, the overall time complexity of the algorithm is $O(m \log m)$. ■

4. Experimental Results

We implemented our algorithm in ANSI C on a Sun SPARC workstation. We tested our algorithm on a number of channel routing benchmark examples from [YoKu82]. The results are summarized in Table 2. Significant reduction on channel density and total net span is achieved¹. (Total net span is the sum of the lengths of the intervals spanned by each net. This value is closely related to the total wire length to route the channel.) The running time of our algorithm is less than one second for these examples. In all cases, the resulting channel density is at most one larger than the greater of the number of left exits and the number of right exits as claimed in Lemma 2. Moreover, we can show that the number of tracks required to route the channel produced by the alternative packing algorithm is the same as the channel density, which is usually not true for arbitrary channels. Fig. 10 and 11 show the channel routing solutions of the example ex3a and the Deutsch's difficult example after pin assignment by our algorithm. Since the left net intervals do not overlap with the right net intervals in the examples in [YoKu82] (which is the case that many heuristics fail), we also tested our algorithm on a number of examples which have many overlaps of the left net intervals and the right net intervals. Our algorithm still performs well on those examples as expected. Fig. 12 shows the solution to an example with many overlaps of left net intervals and right net intervals.

5. Conclusion

In this paper, we present an efficient near-optimal channel pin assignment algorithm for general channels with exits. Such an algorithm has immediate application in solving the combined pin assignment and global routing problem for general cell designs. We are studying several variations of the channel pin assignment problem formulated in this paper. For example, one interesting case is that the positions of the pins on the one edge of the channel are fixed and

¹ Note that most of these examples are from gate array or standard cell design and the pins in these examples may not be permutable. Nevertheless, these examples well serve the purpose of testing the correctness and efficiency of our algorithm.

Examples	Exits		Density		Total net span	
	Left	Right	Before	After	Before	After
ex3a	13	14	15	14	780	194
ex3b	13	10	17	13	820	203
ex3c	13	11	18	13	1126	192
ex4b	0	0	17	2	1346	101
ex5	4	5	20	6	1392	103
ex5b	4	9	20	9	1350	129
Deutsch	1	0	19	2	2539	176

Table 2: Results on the channel routing benchmark examples in [YoKu82].

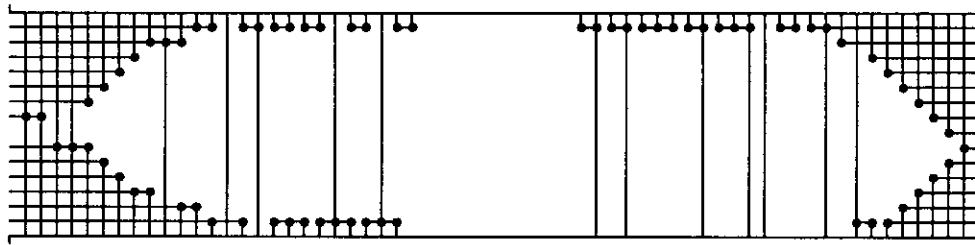


Figure 10: Routing solution of example 3a after pin assignment using our algorithm.



Figure 11: Routing solution to Deutsch's difficult example after pin assignment using our algorithm.

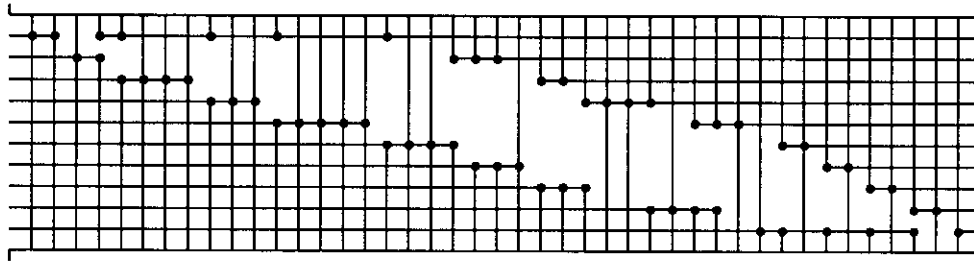


Figure 12: Example of a channel with many exit interval overlaps

we need to determine the positions of the pins on the other edge of the channel. Such an assignment problem arises in solving the combined pin assignment and global routing problem with pre-designed blocks in general cell designs.

6. References

- [BuPe83] Burstein, M. and R. Pelavin, "Hierarchical Channel Router". *Integration, the VLSI journal* (1983) Vol. 1, pp. 21-38.

- [CaWo90] Cai, Y. and D. F. Wong, "An Optimal Channel Pin Assignment Algorithm". *Proc. IEEE Int'l Conf. on Computer-Aided Design* (1990) pp. 10-13.
- [Co89] Cong, J., "Pin Assignment with Global Routing". *Proc. Int'l Conf. on Computer-Aided Design* (Nov., 1989) pp. 294-297.
- [Co91] Cong, J., "Pin Assignment with Global Routing for General Cell Design". *IEEE Trans. on Computer-Aided Design* (Oct. 1991, to appear).
- [De76] Deutsch, D. N., "A Dogleg Channel Router". *Proc. 13th Design Automation Conf.* (1976) pp. 425-433.
- [DeBh89] Deogun, J. and B. Bhattacharya, "Via Minimization in VLSI Routing with Movable Terminals". *IEEE Trans. on Computer-Aided Design* (Aug. 1989) Vol. 8, pp. 917-920.
- [LeLi85] Leong, H. W. and C. L. Liu, "Permutation Channel Routing". *Prof. Int'l Conf. on Computer Design: VLSI in Computers* (1985) pp. 579-584.
- [PeMK90] Pedram, M., M. Marek-Sadowska and E. S. Kuh, "Floorplanning with Pin Assignment". *Proc. IEEE Int'l Conf. on Computer-Aided Design* (1990) pp. 98-101.
- [Pr79] Preas, B., "Placement and Routing Algorithms for Hierarchical Integrated Circuit Layout", Ph. D. Dissertation, Stanford University, 1979.
- [RiFi82] Rivest, R. L. and C. M. Fiduccia, "A 'Greedy' Channel Router". *Proc. 19th Design Automation Conf.* (1982) pp. 418-424.
- [ToTr90] Tollis, I. and S. Tragoudas, "Interchanging Terminals for Improved Channel Routing". *IEEE Int'l Conf. on Circuits and Systems* (May 1990) pp. 344-347.
- [WiWo85] P. Widmayer and C. K. Wong, "An Optimal Algorithm for the Maximum Alignment of Terminals". *Information Processing Letter* (Feb. 1985) Vol. 20, pp. 75-82.
- [YoKu82] Yoshimura, T. and E. S. Kuh, "Efficient Algorithms for Channel Routing". *IEEE Trans. on Computer Aided Design of ICAS* (Jan. 1982) Vol. CAD-1, pp. 25-35.