An Algorithm for Simultaneous Pin Assignment and Routing *

Hua Xiang, Xiaoping Tang, and D.F. Wong
University of Texas at Austin, Austin, TX 78712
{xiangh, tang, wong}@cs.utexas.edu

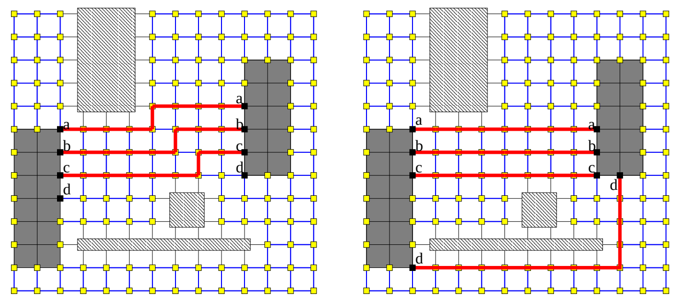
Abstract

Macro-block pin assignment and routing are important tasks in physical design planning. Existing algorithms for these problems can be classified into two categories: 1) a two-step approach where pin assignment is followed by routing, and 2) a net-by-net approach where pin assignment and routing for a single net are performed simultaneously. None of the existing algorithms is "exact" in the sense that the algorithm may fail to route all nets even though a feasible solution exists. This remains to be true even if only 2-pin nets between two blocks are concerned. In this paper, we present the first polynomial-time exact algorithm for simultaneous pin assignment and routing for 2-pin nets from one block (source block) to all other blocks. In addition to finding a feasible solution whenever one exists, it guarantees to find a pin-assignment/routing solution with minimum cost $\alpha \cdot W + \beta \cdot V$, where W is the total wirelength and V is the total number of vias. Our algorithm has various applications: 1) It is suitable in ECO (Engineering Change Order) situations where a designer wants to incrementally modify the existing solution instead of redoing everything after a design change. 2) Given any pin assignment and routing solution obtained by any existing method, our algorithm can be used to increase the number of routed nets and reduce the routing cost. Furthermore, it provides an efficient algorithm for the pin assignment and routing problem of all blocks. The method is applicable to both global and detailed routing with arbitrary routing obstacles on multiple layers. Experimental results demonstrate its efficiency and effectiveness.

1. Introduction

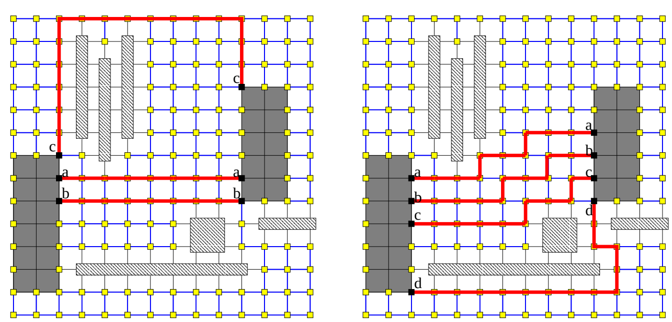
Due to the enormous complexity of VLSI design, a hierarchical approach is needed for the placement and routing of millions of standard cells in order to reduce runtime and improve solution quality. A typical top-down hierarchical approach is as follows: 1) partitioning the circuit into (soft) macro blocks of standard cells, 2) floorplanning the macro blocks (i.e., determining the shapes and locations of the macro blocks), 3) assigning pins on the macro blocks and routing (global/detailed) the nets among the macro blocks, and 4) placing/routing the standard cells within each macro block.

Existing algorithms for macro-block pin assignment and routing can be classified into two categories: 1) a two-step approach where pin assignment is followed by routing [13, 15, 4, 17], and 2) a net-by-net approach where pin assignment and routing for a single net are performed simultaneously [8, 12, 16, 7, 14]. None of the existing algorithms is "exact" in the sense that the algorithm may fail to route all nets even though a feasible solution exists. This remains to be true even if only 2-pin nets between two blocks are concerned. Let us use two examples to illustrate that previous approaches can not guarantee to find a feasible solution. Consider the example in Figure 1. There are 2 macro blocks, 4 nets and 3 obstacles in a single-layer routing environment. In figure 1(a), the pin assignment solution leads to a routing problem that is not routable by any router (at most three nets can be routed). On the other hand, Figure 1(b) shows a feasible (pin assignment and routing) solution. Now consider another example in Figure 2. There are 2 macro blocks, 4 nets and 6 obstacles in a single-layer routing environment. Figure 2(a) shows the result of the net-by-net approach where the net ordering for combined pin assignment and routing is a, b, c, d. As we see, it is not possible to assign and route the pins for net d. Again, Figure 2(b) shows a feasible solution to the problem. (We note that both feasible solutions in Figure 1(b) and Figure 2(b) can be obtained by our algorithm in this paper.)



(a) (b)

Figure 1: 4 nets (a, b, c, d) are to be assigned pins and routed. (a) The two-step approach fails to route all nets (at most 3 nets can be routed for the pin assignment solution). (b) The optimal solution of pin assignment and routing by our approach.



(a) (b)

Figure 2: 4 nets (a, b, c, d) are to be assigned pins and routed. (a) The net-by-net approach fails to route all nets (net d is not routable). (b) The optimal solution of pin assignment and routing by our approach.

*This work was partially supported by the National Science Foundation under grant CCR-9912390.

In this paper, we present the first polynomial-time exact algorithm for simultaneous pin assignment and routing for all 2-pin nets from one block (source block) to all other blocks. In addition to finding a feasible solution whenever one exists, it guarantees to find a pin-assignment/routing solution with minimum cost $\alpha \cdot W + \beta \cdot V$ for any positive pair α and β, where W is the total wirelength and V is the total number of vias. Our algorithm has various applications. 1) It is suitable in ECO (Engineering Change Order) situations where a designer wants to incrementally modify the existing solution instead of redoing everything after a change. 2) Given any pin assignment and routing solution obtained by any existing method, we can increase the number of routed nets and reduce the routing cost by removing the routes connecting to one block and applying our algorithm to redo pin assignment and routing. Furthermore, by applying the algorithm iteratively (each time randomly pick one block as source block), it provides a polynomial-time randomized algorithm for the pin assignment and routing problem among blocks. This method is applicable to both global and detailed routing with arbitrary routing obstacles on multiple layers. Experimental results demonstrate its efficiency and effectiveness.

Our method is based on min-cost flow computations. Although network flow formulations have been proposed for routing in the past [5, 11, 3, 6], there are important differences between our work and previous results. First, previous network flow formulations were primarily designed for global routing, whereas ours combines pin assignment with routing (detailed or global). Second, almost all of those previous works needed to solve the multicommodity flow problem which is NP-hard, whereas our algorithm uses min-cost flow which is a polynomial time solvable problem. Third, our algorithm exactly solves the simultaneous pin assignment and routing problem for all 2-pin nets from one block to all other blocks in polynomial time. (Note that the routing step alone is NP-complete even if there are only two blocks and all nets are 2-pin nets.)

The rest of the paper is organized as follows. Section 2 defines the problem of simultaneous pin assignment and routing in multilayer. In section 3, we give a network flow formulation to find the routes for all 2-pin nets between one block and all other blocks. In section 4, we discuss its application in ECO situation, and demonstrate how to use it to improve any given solution and to solve the pin assignment and routing problem among blocks. Finally, we show the experimental results in section 5 and conclude the paper in section 6.

2. Problem Definition

The macro block layout in multilayer is modeled by a multilayer routing grid graph $G = (V, E)$. It contains not only the topological information, layer and via information, but also the routing obstacle information. So this model is quite flexible and accurate for multilayer technologies, and is suitable for both global routing and detailed routing.

The routing grid graph is 3-dimensional. For convenience, we call the 3 dimensions as x, y, and z dimension. Each layer is an $x - y$ dimensional grid, where x axis goes horizontally and y axis goes vertically. Along z axis, the grid nodes with the same x, y coordinates in different layers are connected by **via** edges. The grid nodes adjacent in the same layer are connected by edges which represent wire segments. In some technology, a layer has a specified track orientation. In this case, if a layer is used for horizontal tracks, horizontally adjacent nodes of the layer are connected by edges.

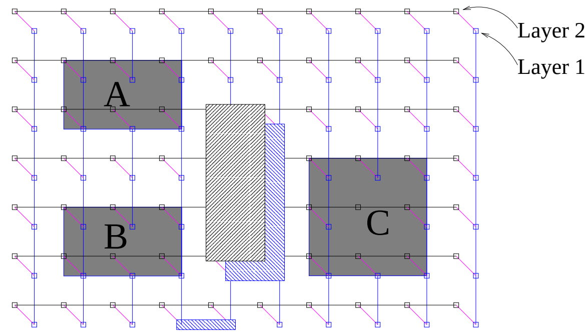


Figure 3: A routing grid graph for two layers. Layer 1 is used for vertical tracks, and Layer 2 for horizontal tracks. Three shaded regions (A,B,C) are macro blocks. The other three regions represent routing obstacles. Blocks occupy Layer 1. There are two obstacles in Layer 1 and one in Layer 2.

Similarly, for layers used for vertical tracks, vertical edges are presented between vertically adjacent nodes. For routing obstacles (routing congestion region, pre-routed wires, crosstalk sensitive region, and so on) where wiring is not allowed, there are no edges (then the nodes inside the obstacles can be omitted). In practice, routing obstacles can present in any layer, which could be modeled by the grid graph. A block can occupy any number of layers. There is no node inside the region that a block occupies, but the nodes on the boundary, which are possible pin locations, are connected to the nodes outside the block. Furthermore, the nodes over a block are present in the graph for over-the-block routing. As an example, Figure 3 illustrates a two-layer routing grid graph, where Layer 1 is used for vertical tracks and Layer 2 for horizontal tracks. Note that there are nodes over blocks for over-the-block routing.

Each edge and each node have a capacity which specifies how many wires are allowed to go through.(In detailed routing, the capacity is 1.) Also each edge is associated with a cost. For wire, the cost is $\alpha \cdot l_e$ where α is specified by users and l_e is the wire length which the edge represents. Notably, for different layers, the cost can be different according to the resistance difference. Accordingly, the cost of edges of different layers should be different. The cost of via edge is β, which can be specified by users.

The goal of pin assignment is to decide the exact pin positions on macro blocks. Routing is to find an appropriate connection among the pins of the same net. The two tasks are closely related. Pin assignment alone neglects many important factors since interconnect is hard to be estimated accurately without carrying out the actual routing step. Moreover, a global view of net information and routing resource information is critical for pin assignment and routing. In this paper, we consider the problem of simultaneous pin assignment and routing for all 2-pin nets between a block (source block) and all other blocks. The general problem of pin assignment and routing among blocks will be discussed in Section 4.

The problem (called **PAR**: Pin Assignment and Routing) is formally described as follows:

Problem 1. **PAR:** *Given a routing grid graph $G = (V, E)$ with U and C where U is a function on edges and nodes denoting the capacity of edges and nodes and C is a function on edges denoting the cost of edges, a set of $m + 1$ macro blocks B(one block is the source block b_s, and the others are sink blocks $b_1, b_2, ..., b_m$), and a set of nets $N =$*

$N_1 \cup N_2 \cup ... \cup N_m$ where $N_i, i = 1, 2, ..., m$ is the set of nets between block b_s and blocks b_i, find a set of paths connecting b_s and $b_1, b_2, ..., b_m$, each path corresponding to a net in N, such that each edge/node is used no more than its capacity and the total cost for all nets is minimized. Each endpoint of a path is a pin location.

In PAR problem, the connections from the points inside the block to the boundary points of the same block are redundant since we can pick the boundary points as pins with less wiring cost. It is true that for multilayer layout a pin can be placed inside a block. However, the pin must be connected with some pin outside the block by over-the-block routing which has to cross the block boundary. Then placing a pin inside a block is unnecessary. Therefore, without loss of generality, we may assume that a pin is only on the boundary of a block and can be placed in any available layer (note that stacked pins are allowed).

As we know, routing itself is a NP-hard problem. It remains NP-hard even if only 2-pin nets between two blocks are concerned. In the traditional two-step approach, the problem is inherently difficult even for two blocks. However, the PAR problem, which combines pin assignment and routing, is solvable in polynomial time. In the following section, we will present an approach to solve the problem optimally.

3. The Algorithm

In this section, we mainly use single-layer routing graph to simplify the presentation, since the single-layer illustration is easier for visualization.

To solve the PAR problem, we first construct a network graph based on the routing graph, and then apply a min-cost flow algorithm [1] to get the solution.

Given a routing grid graph $G = (V, E)$ with capacity U and cost C, blocks $B = \{b_s, b_1, b_2, ..., b_m\}$, and nets $N = N_1 \cup N_2 \cup ... \cup N_m$, the network graph $G_N = (V_N, E_N)$ is constructed as follows.

1. $V_N = \{s, t, t_1, t_2, ..., t_m\} \cup V$, where s is the source node, t is the sink node, t_i is a subsink node.

2. $E_N = E \cup \{(s, v) | v \in P_s\} \cup \{(u_i, t_i) | u_i \in P_i, i = 1, 2, ..., m\} \cup \{(t_i, t) | i = 1, 2, ..., m\}$, where P_s is the set of the available nodes for pin assignment on the boundary of block b_s and P_i is the set of the available nodes on the boundary of block b_i.

3. Edge Capacity: for edge (s, v) and (u, t_i), $U_N(s, v) = U_N(u, t_i) = 1$ in detailed routing and $U_N(s, v) = U_N(u, t_i) =$ pin node capacity in global routing; for edge (t_i, t), $U_N(t_i, t) = |N_i|$; for any other edge $e \in E$, $U_N(e) = U(e)$.

4. Node Capacity: for $v \in V$, $U_N(v) = U(v)$. Other nodes are incapacitated.

5. Cost Function: $C_N(s, v) = 0$, $C_N(u, t_i) = 0$, $C_N(t_i, t) = 0$, for other edge $e \in E$, $C_N(e) = C(e)$.

As an example, Figure 4(b) illustrates the constructed network graph for the PAR problem in Figure 4(a). Note that each undirected edge in G gives a pair of directed edges in opposite directions in G_N.

Note that it is necessary to make nodes capacitated in the network graph G_N (capacitating edges only is not enough since some routes may share the same node without sharing an edge). However, classical network flow problem only

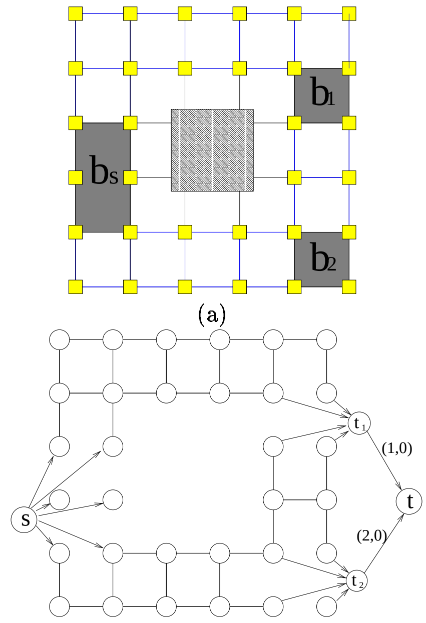


Figure 4: (a) A PAR problem in detailed routing. One net is to be routed between b_s and b_1 and two nets between b_s and b_2. (b) The corresponding network graph. (u, c) specifies the capacity u and cost c of an edge. Each undirected edge represents a pair of directed edges with capacity 1 in opposite directions. All nodes have capacity 1.

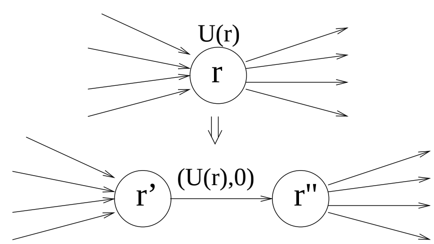


Figure 5: Node splitting for capacitated nodes. The new edge has capacity $U(r)$ and cost 0.

capacitates edges. This can be solved by splitting the capacitated node r into two nodes r' and r'', adding an edge (r', r'') with capacity $U(r', r'') = U(r)$ and cost 0, and turning the original edges (u, r) and (r, v) into edges (u, r') and (r'', v) respectively (refer to Figure 5).

Any flow in the network G_N can be mapped to a pin assignment and routing solution for a subset of the given nets. Figure 6 illustrates a flow f, $|f| = 3$, corresponding to a solution of pin assignment and routing for 3 nets among 3 blocks. Given a set of nets N, let $n = |N|$, i.e., the number of nets in N. If a flow f exists and $|f| = n$, then we can feasibly assign pins and route all nets in N. Furthermore, the cost of the flow is the cost for the solution of pin assignment and routing. Therefore, min-cost flow guarantees a solution with minimum total cost: $\alpha \cdot W + \beta \cdot V$. The total capacities of edges going into sink node t is: $\sum_{i=1}^{m} U_N(t_i, t) = \sum_{i=1}^{m} |N_i| = |N|$. Therefore, the maximum flow f_{max} in G_N, $|f_{max}| \leq |N|$. Then min-cost maximum flow assigns pins and routes for as many nets as possible with minimum total cost.

The following theorem shows that the PAR problem can

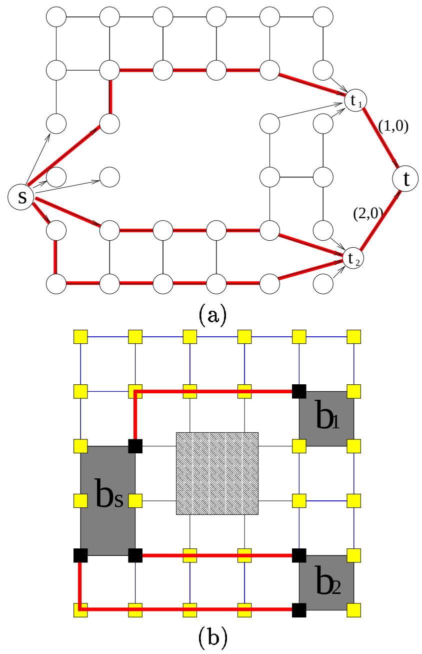


Figure 6: A flow assigns pins and routes for nets. (a) A flow f in the network in Figure 4(b), $|f| = 3$. (b) The corresponding solution of pin assignment and routing for the 3 nets in the problem of Figure 4(a).

be exactly solved by a min-cost flow computation on G_N.

Theorem 1. *A min-cost flow f, $|f| = |N|$, in G_N corresponds to a pin assignment and routing solution to PAR problem for all nets in N with minimum total cost: $\alpha \cdot W + \beta \cdot V$. If the size of the max-flow, $|f_{max}| < |N|$, then there is no feasible solution to the PAR problem, i.e., not all nets in N are routable. A min-cost maximum flow assigns pins and routes for the maximum number of nets with minimum total cost.*

We now summarize the algorithm PAR-by-Flow.

Algorithm PAR-by-Flow(G,U,C,B,N)
1. Construct the network graph $G_N(V_N, E_N)$
2. Assign capacities U_N and costs C_N
3. Apply min-cost maximum flow algorithm on G_N
4. Derive the pin assignment and routing solution

Finding a min-cost maximum flow in a network is a classical problem for which several polynomial-time optimal algorithms are available [9, 2]. Deriving a solution of PAR from a flow in G_N can be done in $O(E)$ time. Thus, if we adopt the double scaling algorithm in [1], we get the following time complexity for the PAR problem.

Theorem 2. *PAR-by-Flow algorithm optimally solves the PAR problem in $O(VE \log\log U_{max} \log(VC_{max}))$ time, for $G = (V, E)$, U_{max} is the maximum value of U, and C_{max} is the maximum value of C.*

Note that the complexity of our algorithm PAR-by-Flow is mainly dependent on the size of the routing grid graph $G = (V, E)$. In global routing model where the size of the routing graph is smaller, PAR-by-Flow requires less runtime.

Computing min-cost flow in a network is independent on the value of costs. This feature is very useful in practice. The wire width and resistance might be different for different layers. The cost for edges in different layers should be adjusted accordingly. In some cases, the vertical and horizontal wire segments of the same layer have different cost, we can assign vertical edges and horizontal edges with different cost too. For example, in a two-layer routing grid where Layer 1 is for vertical tracks and Layer 2 for horizontal tracks, we assign α_v to Layer 1 and α_h to Layer 2 ($\alpha_v \neq \alpha_h$). Then the cost function $(\alpha \cdot W + \beta \cdot V)$ becomes

$$\alpha_v \cdot W_v + \alpha_h \cdot W_h + \beta \cdot V$$

where W_v is the total wirelength for vertical tracks, W_h is the total wirelength for horizontal tracks, and V is the total number of vias. The algorithm guarantees to find a solution with minimum total cost.

In applications, some locations on the boundary of a block may not be allowed for pin assignment. This problem can be solved easily as follows. We remove the directed edge from the source node to the boundary node of the source block which forbids pin assignment, or remove the edge from the boundary node of the sink block to the subsink node. Then our network-flow based algorithm will not assign a pin to the location.

4. Applications

In this section, we discuss applications of PAR-by-Flow algorithm. PAR-by-Flow exactly solves the PAR problem, and can be used as a powerful sub-routine in many situations.

4.1 ECO

PAR-by-Flow provides an optimal solution to pin assignment and routing problem for all 2-pin nets from one block to other blocks. This problem matches well with some situations in ECO (Engineering Change Orders). Usually, a design needs to go through many changes. At each step, designers do not want to redo everything and will just modify the existing solution incrementally. For instance, a designer changes the design of one block in a floorplan. As a result, net connections between the block and other blocks have to be changed accordingly. Some nets become unnecessary; and some new nets need to be added. Also, during re-routing, some routes are kept untouched. Now the problem becomes how to find a new solution subject to these constraints as well as minimizing the total cost $\alpha \cdot W + \beta \cdot V$. The PAR-by-Flow algorithm provides an ideal way to solve this kind of problems. For unchanged routes, we regard them as obstacles. In this way, the pins, wire segments and vias occupied by these nets can not be used by others. Then we update the set of nets according to the added or deleted nets. After removing the connections to the block, we apply PAR-by-Flow and get an optimal solution.

Figure 7 illustrates an example. In Figure 7(a), we have a pin assignment and routing design of a floorplan, and want to change net connections from Block A subject to: (1) keep the routing of 3 nets a, f and g unchanged; (2) delete one net c between Block A and C; (3) add two new nets i and j between A and B. Now we select Block A as the source block. Since the routes for nets a, f, g and the nets among B,C,D and E should not be changed, they are regarded as obstacles. The set of nets becomes $\{b, d, e, h, i, j\}$. The result obtained by our algorithm is shown in Figure 7(b).

4.2 Improvement on Any Given Solution

Our approach not only can be used to improve "local" situation, but also is a good way to improve "global" arrangement. In other words, it can be used to improve any pin

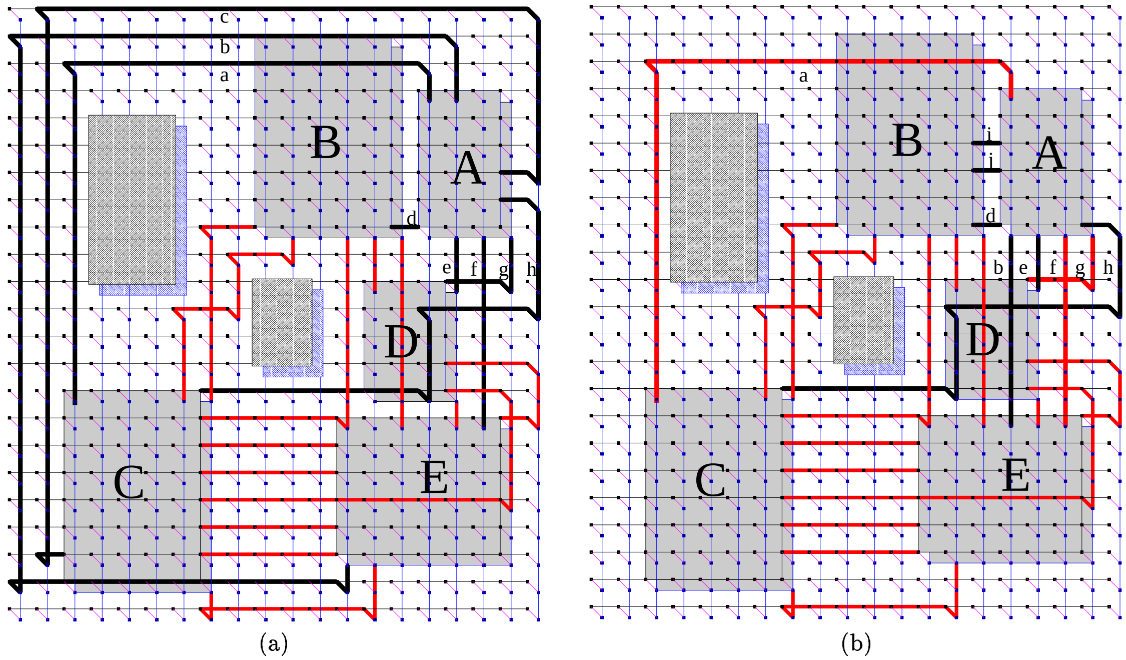


Figure 7: PAR-by-flow in ECO application. (a) The initial pin assignment and routing solution. (b) The solution obtained by applying PAR-by-Flow on Block A. As required, routes a, f and g are unchanged; net c between A and C is deleted, and two new nets i and j are routed between A and B. Note that the cost for net b is greatly reduced as well.

assignment and routing solution. Given a pin-assignment and routing arrangement of blocks, pick a block as the source block and regard others as sink blocks, then remove all routes connected to the source block and apply PAR-by-Flow to redo pin assignment and routing. In the next step, another block is chosen as source block and this process is repeated until all blocks are touched as the source block. The optimality of PAR-by-Flow guarantees that the new solution is no worse than the original one, either increasing the number of routed nets or reducing the cost. By repeating the procedure on each block (as source block), we can improve the solution obtained by any method. We call this iterative method as IMProve-by-PAR.

As we notice, the ordering of source blocks in IMProve-by-PAR has great influence on the final result. Different orderings may lead to very different results since each step is based on the previous step. To alleviate the influence of block order, we implement IMProve-by-PAR by enforcing a random order on blocks to apply PAR-by-Flow. Furthermore, we repeat IMProve-by-PAR for several times to get a better result. Each time, we get a new solution from IMProve-by-PAR, and let this new solution be the input for the next IMProve-by-PAR.

This repeated application of IMProve-by-PAR is referred to as RepIMProve-by-PAR, and its pseudocode is as follows.

Algorithm RepIMProve-by-PAR(G,U,C,N,B,S,T)
S: previous solution; T: iteration number
1. for $i = 1$ to T
2. Generate a random order $Order$ on blocks B
3. S = IMProve-by-PAR(G,U,C,N,B,S, $Order$)
4. endfor

Figure 8 illustrates an example of IMProve-by-PAR. The net connections among 4 blocks are listed in Table 1 and the total number of nets is 17. We use $\alpha = 1$ and $\beta = 1$ in cost function. Figure 8(a) shows an initial solution for pin

Table 1: Net connection (in total 17 nets).

Name	Block A	Block B	Block C	Block D
Block A	0	2	1	4
Block B	2	0	2	4
Block C	1	2	0	4
Block D	4	4	4	0

assignment and routing among 4 blocks. In this solution, only 14 nets are routed and the total cost $W + V = 56$. At the first step, we choose Block A as the source block. After removing all routes connected to Block A, we apply PAR-by-Flow to find net connections for Block A and get a new solution as shown in Figure 8(b). In this step, the cost $W + V$ is reduced by 18. Then we apply PAR-by-Flow to Block B, and two more nets(B-C and B-D) are routed as shown in Figure 8(c). Next, choose Block C as Figure 8(d), and one more net(C-D) is connected, which leads to a complete routing solution. At last, Block D is chosen and nothing is changed. Note that for the final solution, the total cost $W + V$ is reduced from 56 to 45 even though 3 more nets are routed.

In addition, IMProve-by-PAR itself also provides a new way to solve pin assignment and routing problem among multiple macro blocks. The general problem can be decomposed to a set of PAR problems and solved by PAR-by-Flow iteratively. Again, to alleviate the influence of block order, we just choose source block randomly. This comes out a polynomial-time randomized algorithm to solve the pin assignment and routing problem among blocks. Of course, RepIMProve-by-PAR can be used to improve the result. In fact, if we just let the input solutions of RepIMProve-by-PAR S empty, we can always get one solution when the program terminates.

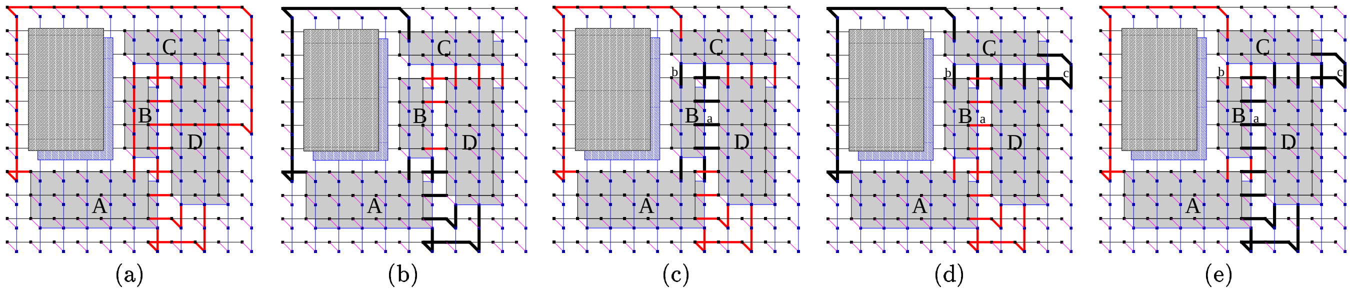


Figure 8: Illustration of improvement on a given solution. (a) Initial net-by-net solution. 3 nets (B-C, B-D, C-D) are not routed. The total cost is 56. (b) The solution after applying PAR-by-Flow on Block A. Cost is reduced by 18. (c) The solution after applying PAR-by-Flow on Block B. Two more nets a and b are routed. (d) The solution after applying PAR-by-Flow on Block C. All nets are routed (3 more nets) with less cost (from 56 to 45). (e) The solution after applying PAR-by-Flow on Block D. Nothing is changed.

5. Experimental Results

We have implemented the PAR-by-Flow and RepIMProve-by-PAR algorithms in C++ language, and carried out experiments on Sun Sparc Ultra 5(360MHz) with 128M memory.

We have tested the refinement algorithm RepIMProve-by-PAR by using a net-by-net approach to get the original solution. The net-by-net approach considers only one net each time, and finds a min-cost path between source block and sink blocks to assign pins and route the net. Nets in net-by-net approach are processed randomly. In RepIMProve-by-PAR, we repeat IMProve-by-PAR 10 times. Both RepIMProve-by-PAR and net-by-net are executed 10 times on 8 data files, 4 for detailed routing and 4 for global routing. Table 2 lists the average of these ten results. For both detailed routing and global routing, after refinement, all nets are routed with a significant improvement on the total cost, the wire length and the number of vias. As an illustration, Figure 9 shows the results of pin assignment and routing for input file "X18", where $\alpha = 1$ and $\beta = 1$. Vertical tracks are in Layer 1 and horizontal tracks are in Layer 2. Figure 9(a) shows the net-by-net solution. Figure 9(b) is obtained by applying our method on (a). The total number of nets is 268. But for net-by-net, 262 nets are routed and the total cost is 11023; after refinement, all nets are connected and the total cost is 7153.

6. Conclusion

In this paper, we have presented the first polynomial-time optimal algorithm for simultaneous pin assignment and routing in multilayer for all 2-pin nets between a source block and all other blocks. Our algorithm is applicable for both global routing and detailed routing with arbitrary routing obstacles on multiple layers, and guarantees a pin-assignment and routing solution with minimum total cost $\alpha \cdot W + \beta \cdot V$ by computing a min-cost flow in a network. In ECO (Engineering Change Order) situation where a designer does not want to redo everything after a change, the algorithm provides an ideal way for incremental modification of the existing solution. Also, it can be applied to improve any pin assignment and routing solution by any existing method. Furthermore, by applying the algorithm iteratively for all blocks (each time randomly pick one block as the source block), it provides a polynomial-time randomized algorithm to solve the pin assignment and routing problem of all blocks. Experiments demonstrate that the algorithm is very efficient and effective.

7. References

[1] R.K. Ahuja, A.V. Goldberg, J.B. Orlin, and R.E. Tarjan. "Finding minimum-cost flows by double scaling", Mathematical Programming 53, pp. 243-266, 1992.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. Network Flows, Prentice Hall, 1993.

[3] C. Albrecht. "Provably good global routing by a new approximation algorithm for multicommodity flow", ACM ISPD-00, pp. 19-25, 2000.

[4] H.N. Brady. "An approach to topological pin assignment", IEEE Trans. Computer-Aided Design, vol. CAD-3, pp. 250-255, 1984.

[5] R.C. Carden and C.K. Cheng. "A global router using an efficient approximate multicommodity multiterminal flow algorithm", ACM/IEEE DAC-91, pp. 316-321, 1991.

[6] J.D. Cho and M. Sarrafzadeh. "Four-bend top-down global routing", IEEE Trans. Computer-Aided Design, vol. 17, pp. 793-802, 1998.

[7] S.G. Choi and C.M. Kyung. "Three-step pin assignment algorithm for building block layout", Electron. Lett., vol. 28, no. 20, pp. 1882-1884, 1992.

[8] J. Cong. "Pin assignment with global routing for general cell designs", IEEE Trans. Computer-Aided Design, vol. 10, pp. 1401-1412, 1991.

[9] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. Introduction to Algorithms, The MIT Press, 1992.

[10] T.C. Hu and M.T. Shing. "The $\alpha - \beta$ routing", VLSI: Circuit Layout Theory, IEEE Press, 1985.

[11] J. Huang, X.L. Hong, C.K. Cheng, and E.S. Kuh. "An efficient timing-driven global routing algorithm", ACM/IEEE DAC-93, pp. 596-600, 1993.

[12] T. Koide, S. Wakabayashi, and N. Yoshida. "An integrated approach to pin assignment and global routing for VLSI building-block layout", Proc. European Conf. Design Automation with the European Event in ASIC Design, pp. 24-28, 1993.

[13] N.L. Koren. "Pin assignment in automated printed circuit board design", Proc. ACM/IEEE Design Automation Conf., pp. 72-79, 1972.

[14] L.E. Liu and C. Sechen. "Multilayer pin assignment for macro cell circuits", IEEE Trans. Computer-Aided Design, vol. 18, pp. 1452-1461, 1999.

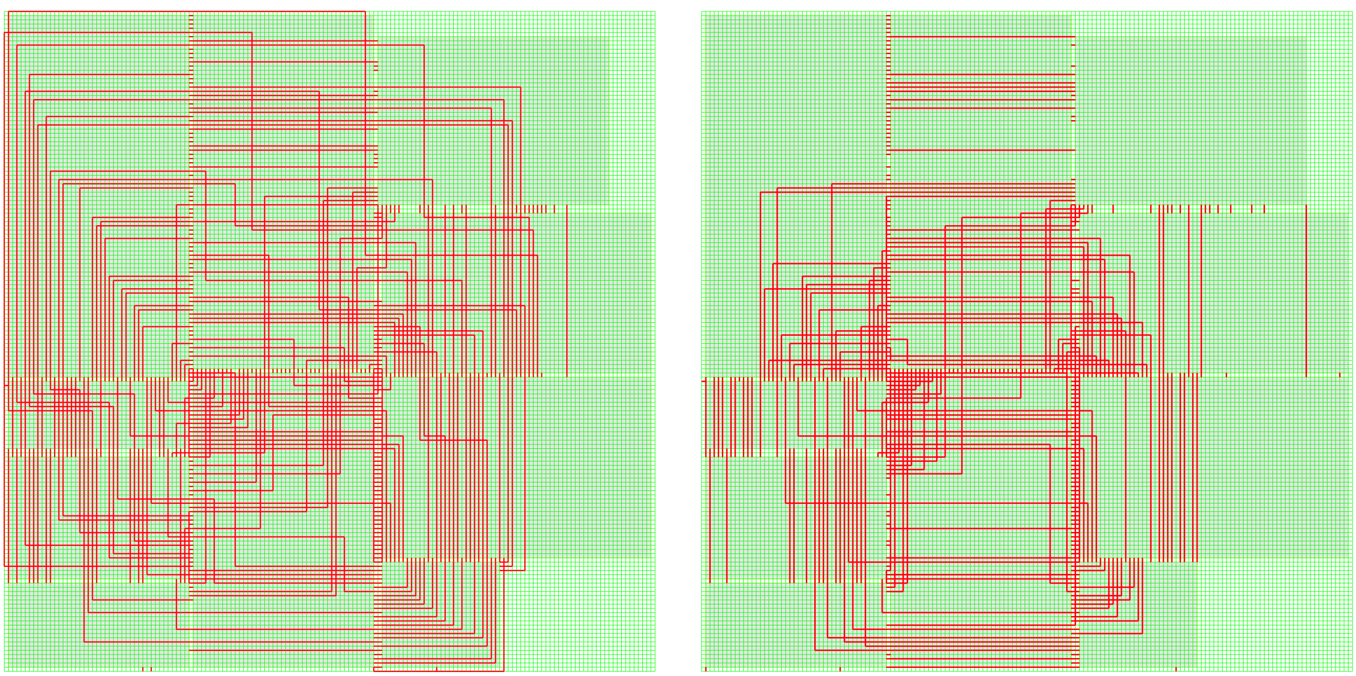
[15] L. Mory-Rauch. "Pin assignment on a printed circuit board", Proc. ACM/IEEE Design Automation Conf., pp. 70-73, 1978.

[16] L.Y. Wang, Y.T. Lai, and B.D. Liu. "Simultaneous pin assignment and global wiring for custom VLSI design", Proc. IEEE Int. Symp. Circuits and Systems, vol. 4, pp. 2128-2131, 1991.

[17] X. Yao, M. Yamada, and C.L. Liu. "A new approach to the pin assignment problem", Proc. ACM/IEEE Design Automation Conf., pp. 566-572, 1988.

Table 2: Average results of RepIMProve-by-PAR for 10 times. All nets are routed after refinement by RepIMProve-by-PAR.

		detailed routing				global routing			
File		C2	Am3	P2	X18	K1	V2	Z2	N3
Grid		67x61	134x140	118x108	155x157	20x21	41x44	33x30	50x47
Block		11	33	12	10	8	20	28	40
Node		8458	38140	25970	49320	944	3824	2168	4980
Edge		29992	125956	86655	157119	3695	14307	8615	18987
Capacity		1	1	1	1	55	30	70	25
Net		152	355	295	268	2131	3088	5498	2588
Time (second)	previous	3.78	35.42	16.44	32.06	5.15	27.52	34.83	43.31
	refined(per iter)	8.94	136.76	43.03	101.72	2.48	26.62	15.57	54.17
Routed nets	previous	150.2	354.5	293.6	263.9	2117.9	3050.1	5440.1	2572.8
	refined	152	355	295	268	2131	3088	5498	2588
Cost	previous (per net)	2575.0 (17.14)	16215.7 (45.74)	6093.8 (20.76)	9817.4 (37.20)	17787.6 (8.40)	52744.8 (17.29)	73015.8 (13.42)	63484.7 (24.68)
	refined (per net)	1966.0 (12.93)	14597.6 (41.12)	5154.5 (17.47)	7191.0 (26.83)	14966.3 (7.02)	48113.8 (15.58)	66722.2 (12.14)	57308.2 (22.14)
	improve	24.6%	10.1%	15.8%	27.9%	16.4%	9.9%	9.5%	10.3%
Wire	previous (per net)	2502.3 (16.66)	15906.8 (44.87)	6012.9 (20.48)	9659.8 (36.60)	16295.7 (7.69)	50782.4 (16.65)	68586.1 (12.61)	60861.0 (23.66)
	refined (per net)	1909.9 (12.57)	14362.7 (40.46)	5098.0 (17.28)	7081.9 (26.43)	13728.9 (6.44)	46611.0 (15.09)	63031.3 (11.46)	55172.4 (21.32)
	improve	24.5%	9.8%	15.6%	27.8%	16.3%	9.4%	9.1%	9.9%
Via	previous (per net)	72.7 (0.48)	308.9 (0.87)	80.9 (0.28)	157.6 (0.60)	1491.9 (0.70)	1962.4 (0.64)	4429.7 (0.81)	2623.7 (1.02)
	refined (per net)	56.1 (0.37)	234.9 (0.66)	56.5 (0.19)	109.1 (0.41)	1237.4 (0.58)	1502.8 (0.49)	3690.9 (0.67)	2135.8 (0.83)
	improve	22.9%	24.1%	32.1%	31.7%	17.1%	23.4%	17.3%	18.6%



(a) (b)

Figure 9: Two-layer pin assignment and routing for X18. (a) Net-by-net solution. Routed nets: 262; cost: 11023. (b) The solution obtained by applying our method on (a). All nets are routed with cost: 7153.