A Faster One-Dimensional Topological Compaction Algorithm

Hsiao-Feng Steven Chen* and D. T. Lee**

Department of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208.

Abstract. We consider the problem of one-dimensional topological compaction with jog insertions. Combining both geometric and graph theoretic approaches we present a faster and simpler algorithm, improving over previous results. The compaction algorithm takes as input a sketch consisting of a set F of features and a set W of wires, and minimizes the horizontal width of the sketch while maintaining its routability. The algorithm consists of the following steps: constructing a horizontal constraint graph, computing all possible jog positions, computing the critical path, relocating the features and reconstructing a new sketch homotopic to the input sketch suitable for detailed routing. The algorithm runs in $O(|F| \cdot |W|)$ worst-case time and space, which is asymptotically optimal in the worst case. Experimental results are also presented.

1 Introduction

Layout compaction is the last stage of VLSI design whose main objective is to minimize the chip area. The algorithm that realizes this task is called a *compactor*[14, 19]. The general 2-D compaction is known to be NP-hard[20], so a number of heuristics were proposed[21, 22, 23], including 1-D compaction, which aims only to minimize horizontal or vertical *width* of the chip[7, 16].

The 1-D compaction problem is generally formulated as a graph-theoretic problem by transforming the layout into a horizontal constraint graph[1] and computing its critical path[11, 23]. The non-horizontal wire segments associated with a net can be treated either as rigid objects(Fig. 1(a)) or flexible (Fig. 1(b)) with bends permitted when the modules are being compacted. In the latter case the compaction is called the *topological compaction*[14] or *homotopic compaction*[17, 18], and the bends are called *jogs*. The best known algorithm for graph-based 1-D compaction (without jog insertions) runs in $O(N^{1.5} \log N)$ time[16], and those for the geometry-based 1-D compaction problem (with jog insertions) run in $O(N^4)$[17] and $O((N^2 + k) \log N)$ time[18] respectively, where N

* Current address: Avant! Corporation, 46871 Bayside Parkway, Fremont, CA 94538, e-mail: schen@avanticorp.com.

** Supported in part by the National Science Foundation under the Grant CCR-9309743, and by the Office of Naval Research under the Grants No. N00014-97-1-0514, and No. N00014-95-1-1007. e-mail:dtlee@ece.nwu.edu.

[1] From now on we assume that the horizontal width of the chip is to be minimized.

is the input size, and k is some measure of the differences between the input and output topology which may be $O(N^3)$. Gao et al. also proposed an $O(N^2 \log N)$ time algorithm for the topological compaction with a non-trivial implementation for both planar and knock-knee routing model[8]. Although the topological compaction algorithms are proposed for only one-layer routing model, extensions to multi-layer layout model can be done by applying these algorithms to every layer at the same time[18]. Moreover, it could also be applied to general-architecture layout style used in sea-of-gate chips[1, 3, 9, 15].

Generally speaking, the basic steps of a topological compactor are as follows: Firstly the layout topology is described by a *sketch* model consisting of *features* or *modules* and flexible *wires*. The wires are usually transformed into a normal form, called *rubber-band equivalent*(RBE)[2, 4, 5, 6, 13]. Secondly the moving ordering and the displacement of the modules are determined, while their relative positions, and the so-called *safe* routing condition for wires[4, 13] are maintained. Finally a *homotopic router* is employed to reconstruct the detailed routing for the nets according to the *homotopy*[4, 13] and the required routing model, i.e., single-layer[13] in grids or gridless, knock-knee model[10], etc. We present a faster and simpler algorithm that runs in $O(|F| \cdot |W|)$ time for the 1-D topological compaction problem, where $|F| + |W|$ reflects the input size. Some experimental results will be given at the end.

2 Preliminaries

The input to a homotopic router or a topological compactor is represented by a *sketch*[13, 18]. A sketch $S = (F, M, W)$ is a planar drawing of a physical layout in the grid, where F is a finite set of *features*, which are points or open line segments on the boundaries of the modules m_i's in M, and W is a finite set of *wires* or *nets*, which are vertex-disjoint simple paths connecting the point features by a sequence of straight line segments (Fig. 2). We assume each net has two endpoints, called *terminals*, and all the features lie on a grid.

A detailed routing of sketch $S(F, M, W)$, $W = \{\mu^1, ..., \mu^n\}$, is a sketch $S^*(F, M, W^*)$, $W^* = \{\nu^1, ..., \nu^n\}$, of minimum wire length such that ν^i is *homotopic* to μ^i, i.e., net μ^i and net ν^i have the same pair of terminals, and μ^i can be transformed continuously into ν^i without intersecting any feature in F, and ν^i's in W^* lie on a rectangular grid of unit-spacing, where n is the number of nets. Fig. 2(b), (c) and (d) show respectively *vertex-disjoint RBE* (VRBE[2] for short) $\overline{S}$, a detailed routing S^* and a topologically compacted sketch S^c of sketch S in Fig. 2(a). Note that with conventional 1-D compaction, the sketch S can only be compacted by one unit (Fig. 2(c)), while with topological compaction it can be compacted by three units (Fig. 2(d)).

The homotopy of a net μ^j in W is specified by a sequence of wire segments, denoted as $\mu^j =< w_0^j, w_1^j, ..., w_l^j >$ for some l, where $w_k^j = \overline{s_k^j, s_{k+1}^j}$ is a line

[2] A vertex-disjoint *RBE* is an *RBE* such that all the polygonal shortest paths homotopic to the nets are vertex-disjoint on the grid.

segment. The size of the wire homotopy is the total number of wire segments, denoted as $|W| = \sum_j \sum_k w_k^j$.

Lemma 1 [2] *Given a sketch $S(F, M, W)$, its vertex-disjoint rubber-band equivalent sketch can be computed in time $O(|F| \cdot |W|)$ in the worst case.*

Let $c = (x(c), y(c))$ and $d = (x(d), y(d))$ be two points in F, and let $\overline{cd}$ denote the open line segment between c and d. $\overline{cd}$ is a *cut* if it intersects no features in F. A cut $\overline{cd}$ is *critical* if either (1) both c and d are point features, or (2) c is a point feature, and d is the point on a line feature closest to c. The *capacity* of a cut is defined as the *maximum* number of legal wires that can cross the cut. The actual number of wires crossing a cut is called the *density* of the cut. A cut is said to be *safe*, if its density is not greater than its capacity. It has been shown that a sketch $S(F, M, W)$ is routable if and only if every critical cut is safe[4, 13].

A *diagonal cut* $\overline{cd}$ is a cut emitted from point or corner feature c to d with slope $+1$ or -1. Associated with each point feature f_i there are at most four diagonal cuts denoted $f_i(\theta)$, where $\theta = k\pi/4, k = 1, 3, 5, 7$, denotes the angle formed by the cut and the positive x-axis. To obtain a routable sketch the precise distance of wires from corner features, against which a number of paths resulting from the VRBE transformation are tightened, must be computed. Suppose w is the kth wire touching against a corner feature f. Then w must cross a diagonal cut $f(\theta)$ at a point z whose distance from f is at least $|\overline{fz}| = \sqrt{2}k$. The segment $\overline{fz}$ with respect to wire segment w is called an *obstacle spoke* for w, and z is a *supporting vertex* of w. A spoke is positive if its slope is $+1$, and negative if -1.

In [2] Chen and Lee proposed an algorithm to transform a sketch into a VRBE and provided a new routability condition: given a sketch $S(F, M, W)$, an unsafe critical cut exists if and only if there exists an illegal vertex in some net $\mathcal{N}^j \in \overline{W}$ of its VRBE $\overline{S}(F, M, \overline{W})$ or there exist intersecting paths (Fig. 3). An internal vertex of $\mathcal{N}^j$ in $\overline{W}$ is *illegal* if the angle formed by $\mathcal{N}^j$ and the obstacle spoke supporting the VRBE is less than $\pi/4$.

Denote the displacement of a feature $f \in F$ by $\mathbf{d}(f)$ and call the vector $\mathbf{d} \in \mathbf{R}^{|F|}$ of displacements a *configuration*. To maintain the same topology as in the given sketch S, the following two conditions must hold: (1) $x(p) + \mathbf{d}(f) < x(q) + \mathbf{d}(g)$ for any $p = (x(p), y(p))$ on feature f and and $q = (x(q), y(q))$ on feature g such that $x(p) < x(q)$ and $y(p) = y(q)$, and (2) $\mathbf{d}(f) = \mathbf{d}(g)$ for any two features in the same module. The compacted sketch is denoted by $S(\mathbf{d})(F(\mathbf{d}), M, W^c)$ such that a point p on feature f with coordinates $(x(p), y(p))$ in S has coordinates $(x(p) + \mathbf{d}(f), y(p))$ in $S(\mathbf{d})$ and the wires W^c have the same global routes as W in S. Let $\mathcal{C}(S) \subseteq \mathbf{R}^{|F|}$ be the *configuration space* that satisfies the above conditions. The topological compaction problem can be formalized as the following:

> Given a routable sketch $S(F, M, W)$, find a configuration $\mathbf{d} \in \mathcal{C}(S)$ such that $S(\mathbf{d})$ has minimal x-width.

Theorem 1 *The one-dimensional topological compaction problem can be solved in time $O(|F| \cdot |W|)$, which is asymptotically optimal in the worst case.*

3 Horizontal Constraint Graph and Jog Insertion

The horizontal constraint graph $G(V, E)$, specifying the minimal horizontal spacing requirements between the left and right vertical edges of any two horizontally visible rigid objects, is constructed as follows: Given a sketch, the left or right vertical edge of each object (a feature or a wire segment) is mapped to a vertex $v \in V(G)$. The arcs in $E(G)$ represent the horizontal visibility from left to right between the objects and the weight of each arc connecting two vertices specifies the minimum spacing requirement between the two objects represented by the vertices. A plane sweep method can be performed to compute the horizontal visibility between the objects and G with $O(N)$ vertices and edges in $O(N \log N)$ time, where $N = |F| + |W|$. Assume there are two special vertical line pseudo features $\mathbf{L}$ and $\mathbf{R}$ positioned as the *leftmost* and *rightmost* rigid objects and therefore two nodes also named $\mathbf{L}$ and $\mathbf{R}$ in G. The minimal width between them can be obtained by applying a critical path algorithm to G. In particular, each vertex v is associated with a field $x[v]$ defined by $x[v] = \max_{u \in \pi(v)}\{x[u] + w(< u, v >)\}$, where $w(< u, v >)$ is the weight of the arc $< u, v >$ and $\pi(v)$ is the set of vertices connected to vertex v. $x[v]$ represents the minimum legal distance between the source vertex $\mathbf{L}$ and v. The sink vertex $\mathbf{R}$ will eventually receive a value $v[\mathbf{R}]$ that is the minimum width.

Lemma 2 *The horizontal constraint graph $G(V, E)$ associated with a sketch S of N objects $(N = |F| + |W|)$ can be computed in time $O(N \log N)$. Moreover, the longest path can be computed by a topological sort in time $O(N)$.*

We now consider topological compaction. The wire segments $w_k^j = (s_k^j, s_{k+1}^j) = ((x_k^j, y_k^j), (x_{k+1}^j, y_{k+1}^j))$ in each net $\mathcal{N}^j \in \overline{W}$ in VRBE $\overline{S}(F, M, \overline{W})$ will be partitioned into two subsets, H^j and V^j, consisting of horizontal segments and non-horizontal segments respectively. The non-horizontal wire segments are further decomposed into $\overline{C^j}$, a collection of *y-monotone chains* $\mathcal{P}_t^j$ for some t. Let $\overline{C}$ denote the collection of $\overline{C^j}$ and $\overline{H}$ the collection of $\overline{H^j}$ for all net $\mathcal{N}^j, 1 \leq j \leq n$. That is, $\overline{W} = \overline{H} \cup \overline{C}$: $\overline{H} = \{w_k^j : |y_k^j - y_{k+1}^j| = 0,\ 0 \leq k < l\}$ and $\overline{C} = \{\mathcal{P}_t^j : |y_k^j - y_{k+1}^j| \neq 0,\ 0 \leq \alpha \leq k < \beta \leq l\}$, where $\mathcal{P}_t^j = (w_\alpha^j, \ldots, w_\beta^j)$ and y_α^j and y_β^j are the minimal and maximal y-coordinates, respectively over y_k^j. Fig. 4(a) shows y-monotone chains $\overline{C}$ and horizontal wire portions $\overline{H}$ in bold and shaded lines respectively. $\mathcal{N}^j$ has three y-monotone chains, i.e., $t = |\overline{C^j}| = 3$.

Lemma 3 *Given a sketch $S(F, M, W)$, the total number of y-monotone chains in VRBE $\overline{S}$, denoted $|\overline{C}|$, satisfies $|\overline{C}| \leq |W|$.*

Obviously, a jog can only be introduced in the y-monotone chains. To highlight this visibility relation between the features and chains, and to ease the description of our algorithm, we introduce a *chain sketch* derived from VRBE $\overline{S}$ as follows: A chain sketch $\overline{\overline{S}}(\mathbf{d}) = (\overline{F}(\mathbf{d}), M, \overline{\overline{C}})$ is a routable sketch transformed from VRBE sketch $\overline{S}(F, M, \overline{C})$ by some $\mathbf{d} \in \mathcal{C}(S)$ such that $\overline{\overline{S}}$ has the same (horizontal) visibility as in $\overline{S}$ but every y-monotone chain $\mathcal{P}_t^j \in \overline{C}$ is replaced with a vertical line segment $I_t^j(x) = (x, y_\alpha^j, y_\beta^j) \in \overline{\overline{C}}$ for some x and the horizontal distances between every pair of objects are legal. We call $I_t^j(x)$ the *chain interval* corresponding to y-monotone chain $\mathcal{P}_t^j$. Fig. 4(b) shows the associated chain sketch $\overline{\overline{S}}$ for the VRBE sketch in (a). Note that in $\overline{\overline{S}}$, the y-coordinates y_α^j and y_β^j for each $I_t^j(x)$ are equal to those of chain $\mathcal{P}_t^j$ and the relative visibility information from VRBE $\overline{S}$ is maintained. Since $\overline{C}$ to $\overline{\overline{C}}$ is a one-to-one mapping, $|\overline{C}| = |\overline{\overline{C}}|$. The x-coordinate of $I_t^j(x)$ can be determined by $x[j]$ of vertex j in the constraint graph $\overline{G}$ obtained from VRBE $\overline{S}$. We omit the description of the computations of $\overline{G}$ and of chain sketch $\overline{\overline{S}}$.

Consider Fig. 5 and its constraint graph. Fig. 5(a) shows that if we can somehow identify the position(s) to split the vertices associated with the y-monotone chains, further compaction is possible by breaking one longer critical path into two shorter ones. As shown in Fig. 5(b), the original longest path $\mathbf{L} \rightarrow a \rightarrow b \rightarrow c \rightarrow d \rightarrow \mathbf{R}$ is divided into two, i.e., $\mathbf{L} \rightarrow a \rightarrow b \rightarrow c \rightarrow \mathbf{R}$ and $\mathbf{L} \rightarrow b' \rightarrow c' \rightarrow d \rightarrow \mathbf{R}$, by splitting off new vertices b' and c' from the original vertices b and c. Fig. 5(c) shows the final compacted result. Note that if we move a or d one unit closer, the wires will not be allowed to bend, for otherwise the layout would become illegal. In our graph-theoretic approach for topological compaction, such a vertex splitting is a crucial step. Our main contribution is to provide a systematic method to introduce jogs breaking the longest path, thereby obtaining an optimal width.

4 Jog Vertex Determination

Consider an example (Fig. 6) containing a point feature f with an obstacle spoke f^1 of length $\sqrt{2}$ in the direction $\theta = 3\pi/4$ and three nets. Imagine moving f toward left. f^1 will hit the first net at vertex z^1 as indicated. The spoke f^1 is now translated to become spoke f^2. Continuing the movement of f toward the left, the first net will be bent and f^2 will hit a vertex z^2 on the second net. In so doing we obtain spokes f^1, f^2, f^3 and f^4 and vertices z^1, z^2 and z^3. This process terminates when spoke f^4 hits a line feature at vertex z^4. The vertices z^i on the nets are the *potential supporting vertices*, or *jog vertices*, at which the nets can be bent, when the features are compacted. The spokes f^1 through f^4 are referred to as P-spokes. For a formal definition of P-spokes see the full version of this paper.[3] The lemma below follows immediately from Lemma 3.

Lemma 4 *Given a sketch $S(F, M, W)$, let J denote the set of induced jog vertices and $\overline{\overline{C}}$ the set of chain intervals in the associated chain sketch $\overline{\overline{S}}$. Then we have (1) each chain interval has at most $|F|$ jog vertices and (2)* $|J| \leq \sum_{i=1}^{|\overline{\overline{C}}|} |F| = |F| \cdot |\overline{\overline{C}}| \leq |F| \cdot |W|$.

[3] H.-F.S. Chen and D.T. Lee, "A faster one-dimensional topological compaction algorithm with Jog Insertion," manuscript, submitted for publication, 1997.

To determine the actual jog positions, we make the following observations. If a sketch after compaction is unroutable, it must be caused by the existence of illegal vertices which results from some overpushed features. Moreover, the illegal vertices must occur in pairs on the same net by two P-spokes in opposite horizontal directions (one toward left and one toward right) (see Fig. 7: if the two features move any closer, the two jog vertices become illegal). Consider two jog vertices A and B, $y(A) < y(B)$, along some net $\mathcal{N}^j$ next to each other that are induced by spokes $f_2^{k'}(\theta)$ and $f_1^k(\theta + \pi)$ with $\theta = \frac{3\pi}{4}$ or $\frac{\pi}{4}$, respectively, as shown in Fig. 7. There are two possible constraint types between f_1 and f_2:

- $\lambda(\frac{3\pi}{4}, \frac{7\pi}{4})$ or λ_--constraint for brevity: Formed by the spokes of negative slope, $f_2^{k'}(\frac{3\pi}{4})$ and $f_1^k(\frac{7\pi}{4})$ (Fig. 7(a)).
- $\lambda(\frac{\pi}{4}, \frac{5\pi}{4})$ or λ_+-constraint for brevity: Formed by the spokes of positive slope, $f_2^{k'}(\frac{\pi}{4})$ and $f_1^k(\frac{5\pi}{4})$ (Fig. 7(b)).

These two types of constraint will be captured by the constraint graph $\overline{G}$. Since A and B are induced by P-spokes of features f_1 and f_2, there must exist a path in $\overline{G}$ between the two nodes associated with f_1 and f_2 passing through a node $\overline{AB}$, as shown in the rightmost graphs in Fig. 7. We have:

Lemma 5 *A routable sketch S compacted with a graph-theoretic 1-D compactor is still routable.*

On the other hand, for jog vertices A and B defined above, where $y(A) \geq y(B)$, there is no horizontal spacing constraint between features f_1 and f_2 since no matter how the features are moved, jog vertices A and B are always legal. That is, the horizontal constraint between f_1 and f_2 can be *removed* in this case. This is done by splitting the vertex corresponding to $\overline{AB}$. As a result of this node splitting (Fig. 8), all the incoming edges and outgoing edges to and from v should all be partitioned accordingly. Assume that the y-monotone chain c gets split at vertex B, and nodes v_1 and v_2 correspond to the portions of the y-monotone chain above and below B respectively. Then those features *visible* from the portion of c above B will be connected to v_1 and those *visible* from the portion of c below B will be connected to v_2. This pair of P-spokes created by features f_1 and f_2 is said to form a λ_+-free and λ_--free configuration, depending on the signs of the slopes of the pair of spokes respectively. These λ-free configurations provide the necessary and sufficient conditions to locate effective jogs. The modified horizontal constraint graph is called *decomposed constraint graph.* Note that some λ-constraints are not *real* and can be ignored.

Lemma 6 *A routable sketch compacted with the decomposed constraint graph due to the λ-free configuration is still routable.*

Lemma 7 *Features f_i and f_j form a false λ-constraint if and only if on the same net $\mathcal{N}^j$, f_i and f_j form (1) a λ_--constraint and a λ_+-free configuration, or (2) a λ_+-constraint and a λ_--free configuration.*

Due to space limit we illustrate this with an example. Fig. 9(a) shows a *false* λ_+-constraint, because after the constraint graph is split at point J, there is no path between features f_i and f_j in the constraint graph, while Fig. 9(b) shows a *real* λ-constraint.

For each λ_--free configuration one can introduce a jog but this approach will give far more jogs than necessary. It turns out that some of the λ-free configurations are redundant. In general each λ-free configuration can be represented as an interval, called *free interval*. A free interval is *redundant* if it contains another free interval in its interior, and is *nonredundant*, otherwise. The jog introduced for a nonredundant interval will serve the purpose for all free intervals containing the nonredundant interval. Finding the minimum number of of jogs is equivalent to finding a minimum clique cover for interval graphs, and can be done in linear time if the endpoints of the intervals have been sorted[12].

5 Topological Compaction Algorithm and Its Complexity

The proposed compaction algorithm combines both graph-theoretic and geometric approaches. The first step in our algorithm is a VRBE preprocessing on the input sketch. We then focus on the y-monotone chains by transforming them to chain sketch $\overline{\overline{S}}$. This step is done in conjuction with the computation of a horizontal constraint graph G using the technique described in Section 3 for $\overline{S}$. That is, after applying the critical path algorithm to G, we obtain the x-coordinate value of each chain interval and displacement of point and line features. The graph resulting from the chain sketch $\overline{\overline{S}}$, is the initial constraint graph. To cover all potential jogs we create the jog vertices following the λ-free configurations. The collection of the jog vertices is stored in homotopy W^1 in the new sketch S^1. In particular, if a λ-free configuration is detected, the chain in $\overline{\overline{S}}$ will be split into two subchains and new nodes and arcs generated in the constraint graph. The new graph obtained is the decomposed constraint graph G_m associated with G. Once this graph is found, by computing its critical path, all the vertices for the features possess the information about how much they can be shifted toward the source vertex. Therefore, the new (and correct) chains provide the detailed routing. The following algorithm (with details omitted) summarizes the above steps.

Algorithm 1D-Topological-Compaction
Input : A sketch $S = (F, M, W)$.
Output: Width minimizing sketch S^c, in which W^c is VRBE or detailed routing conforming to the original homotopy W.
Method:

1. $\overline{S}(F, M, \overline{W}) \leftarrow$ VRBE_Transformation(S)
2. $\overline{C} \leftarrow$ Chain_Interval($\overline{W}$)
3. $\overline{\overline{S}(\mathbf{d})} = (F', M, \overline{\overline{C}}) \leftarrow$ Chain_Transformation $(\overline{S}(F, M, \overline{C}))$:
   - $G \leftarrow$ Horizontal_Constraint_Graph($\overline{S}$)
   - $\mathbf{d}$ = Compute_Critical_Path(G) ($\mathbf{d}= (\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_{|F|})$)

– $F' \leftarrow F(\mathbf{d})$.
– Deterimine the x-position of the chain intervals in $\overline{\overline{C}}$ based on $\mathbf{d}$.

4. $S^1(F', M, W^1) \leftarrow$ Compute_Jog_Vertex($\overline{\overline{S}}$)
5. $\overline{\overline{S}} \leftarrow$ Split($\overline{\overline{S}}, W^1$)
6. $G_m \leftarrow$ Decompose($G, \overline{\overline{S}}$)
7. $\mathbf{d}_m =$ Compute_Critical_Path(G_m)
8. $F'' \leftarrow F(\mathbf{d}_m)$
9. $S^c(F'', M, W^c) \leftarrow$ Reconstruct_VRBE(S^1) (or output the detailed routing formed by $\overline{\overline{S}}$)
10. **return** S^c

The VRBE preprocessing takes time $O(|F| \cdot |W|)$ (Lemma 1). To construct the horizontal constraint graph G, the algorithm visits each of the jog vertices so its time complexity is $O(N \log N + |\overline{W}|) = O(|F| \cdot |W|)$. The step Compute_Jog_Vertex() takes time $O(N_1 \log N_1 + |F| \cdot |W|)$ for both scanning and reporting for all the jog vertices, $N_1 = O(|F| + |W|)$). To construct the horizontal constraint graph G_m, the algorithm visits each of the jog vertices so its time complexity is also $O(|F| \cdot |W|)$. Finding the critical path is proportional to the size of the acyclic digraph. Since the number of arcs may be at most the number of the jog vertices, the time complexity again is $O(|F| \cdot |W|)$. Finally, shifting the features and wires takes $O(|F| + |F| \cdot |W|)$. Therefore, the overall running time of the compaction algorithm is $O(|F| \cdot |W|)$. Based on the lower bound argument for a sketch model given in [13], which is $\Omega(|F| \cdot |W|)$, the overall time complexity is asymptotically optimal in the worst case. This complete the proof of **Theorem 1**.

6 Conclusion

In this paper we have presented an efficient and simpler single-layer 1-D topological compaction algorithm that improves over previous results [8, 17, 18]. The proposed algorithm can be extended to multi-layer technology and generalizes easily to applications like cell-level layout, macro-level, building block layout, or general-architecture layout etc.

The algorithm was implemented in C on a geometric input/output device, called GeoSheet, developed at Northwestern University. Fig. 10 and Fig. 11 show the computer output of two test examples in which diagram (a) gives the input sketch, (b) gives the result after conventional 1-D compaction without modules, (c) gives the result after 1-D compaction with modules, (d) gives the result after the proposed topological compaction without modules, and (e) gives the result after topological compaction with modules. (In fact, (b) and (c) are the chain sketch $\overline{\overline{S}}$ corresponding to the initial sketch S in (a).)

References

1. R.C. Carden IV and C.K. Cheng, "A global router using an efficient approximate multicommodity multiterminal flow algorithm", *Proc. 28th DAC*, 316-321, 1991.

2. H.-F.S. Chen and D.T. Lee, "A faster algorithm for rubber-band equivalent transformation for planar VLSI layouts," *IEEE Trans. Computer-Aided Design*, 217-227, Feb. 1996.
3. C. Chiang, M. Sarrafzadeh, and C.K. Wong, "Global routing based on Steiner minmax trees," *IEEE Trans. Computer-Aided Design* 1318-1325, Dec. 1990.
4. R. Cole and A. Siegel, "River routing every which way, but loose," *Proc. 25th Ann. Symp. on Foundations of Comp. Sci.*, 65-73, Oct. 1984.
5. W.W.-M Dai, R. Kong, and J. Jue, "Rubber band routing and dynamic data representation," *Proc. ICCAD*, 1990, pp.52-55.
6. W. W.-M. Dai, T. Dayan and D. Staepelaere, "Topological routing in SURF: Generating a rubber-band sketch," *Proc. of the 28th DAC*, 39-44, 1991.
7. A.E. Dunlop, "SLIP: Symbolic layout of integrated circuits with compaction," *IEEE Trans. Computer-Aided Design*, Nov. 1978.
8. S. Gao and F.M. Maley, "Advances in homotopic layout compaction," *Proc. of the 1989 Symp. on Parallel Alg. and Arch.*, 273-282, 1989.
9. W.-C. Kao, and T.-M. Parng, "Cross point assignment with global rerouting for general-architecture designs," *IEEE Trans. Computer-Aided Design*, 337-348, March, 1995.
10. M. Kaufmann and K. Mehlhorn, "A linear time homotopic routing algorithm," *SIAM on Computing*, 227-246, April, 1994.
11. G. Kedem and H. Watanbe, "Graph-optimization techniques for IC-layout and compaction," *IEEE Trans. Computer-Aided Design*, 12-20, 1984.
12. D.T. Lee, M. Sarrafzadeh and Y.F. Wu, "Minimum cut for circular-arc graphs," *SIAM J. Comput.,* 19,6 1041-1050, Dec. 1990.
13. C.E. Leiserson and F.M. Maley, "Algorithms for routing and testing routability of planar VLSI layouts," *Proc. 17th Annu. ACM STOC*, 69-78, 1985.
14. T. Lengauer, Combinatorial Algorithms for Integrated Circuit Layout, John Wiley & Sons, 1990.
15. J.T. Li and M. Marek-Sadowski, "Global routing for gatearray," *IEEE Trans. Computer-Aided Design*. 298-307, Oct., 1984.
16. C.-Y. Lo, and R. Varadarajan, "An $O(n^{1.5} \log n)$ 1-D compaction algorithm," *Proc. of the 26th DAC.*, 382-387, 1990.
17. F.M. Maley, "A generic algorithm for one-dimensional homotopic compaction," *Algorithmica*, 103-128, 1991.
18. K. Mehlhorn and S. Näher, "A faster compaction algorithm with automatic jog insertion," *IEEE Trans. on Computer-Aided Design*, 451-466, Feb. 1990.
19. T. Ohtsuki, Layout Design and Verification, Advances in CAD for VLSI, vol. 4, North-Holland, 1986.
20. S. Sastry and A. Parker, "The complexity of 2-dimensional compaction of *VLSI* layouts," *Proc. IEEE Int'l Conf. on Circuits & Computers*, 402-406, 1982.
21. M. Schlag, Y.Z. Liao and C.K. Wong, "An algorithm for optimal two-dimensional compaction of VLSI layouts," *Integration, VLSI journal*, 179-209, 1983.
22. H. Shin and C.-Y. Lo, "An efficient two dimensional compaction of VLSI layouts," *Proc. of 26th DAC.*, 290-295, 1990.
23. W.H. Wolf, R.G. Mathews, etl., "Algorithms for optimizing two dimensional symbolic layout compaction," *IEEE Trans. on Computer-Aided Design*, 451-466, Apr. 1988.

This article was processed using the LaTeX macro package with LLNCS style

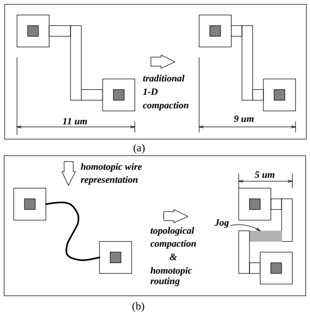

Fig. 1. Traditional 1-D compaction versus topological compaction

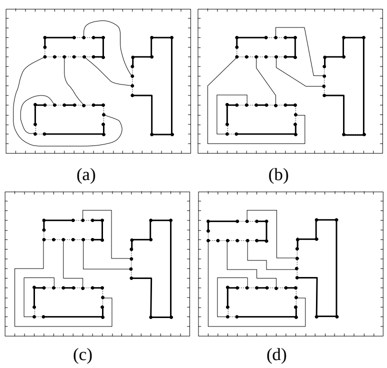

Fig. 2. Sketches (a) S (b) $\overline{S}$ (c) S^* (d) S^c

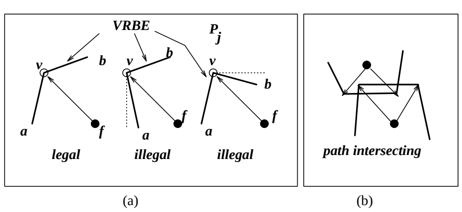

Fig. 3. Two unsafe conditions in vertex-disjoint RBE.

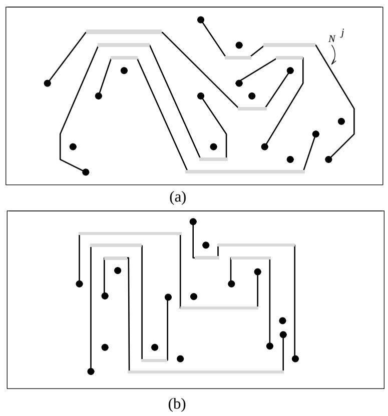

Fig. 4. (a) y-monotone chains in a VRBE sketch (bold lines) (b) the chain sketch $\overline{\overline{S}}$

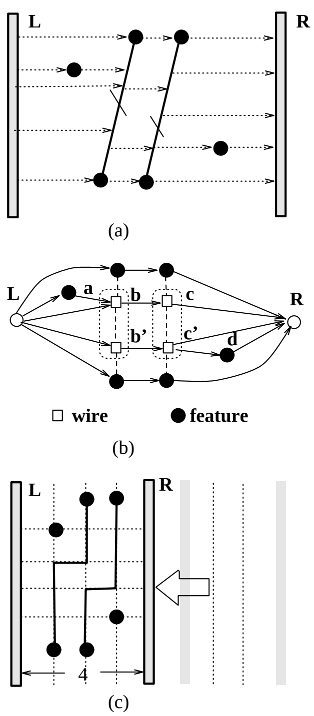

Fig. 5. (a) A sketch (b) constraint graph (c) after compaction.

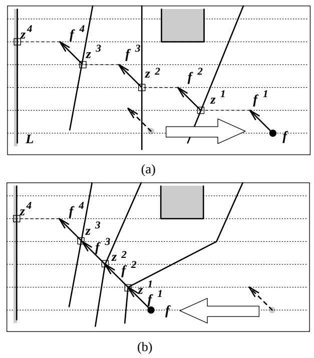

Fig. 6. P-spokes in $3\pi/4$ direction

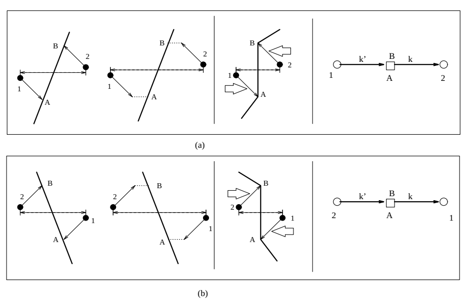

Fig. 7. (a) Constraint type $\lambda(7\pi/4, 3\pi/4)$ and (b) $\lambda(\pi/4, 5\pi/4)$

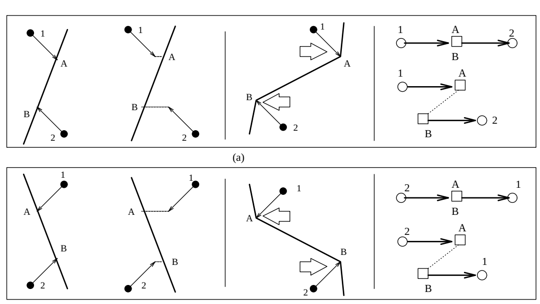

Fig. 8. (a) λ_{-}-free and (b) λ_{+}-free configurations

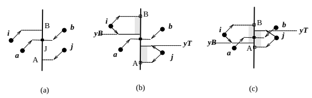

Fig. 9. (a) Overlapping (b) False constraint (Case 1) (c) Real constraint (Case 2)

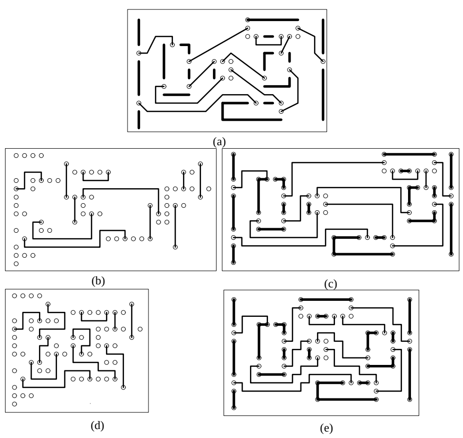

Fig. 10. Test example S_1.

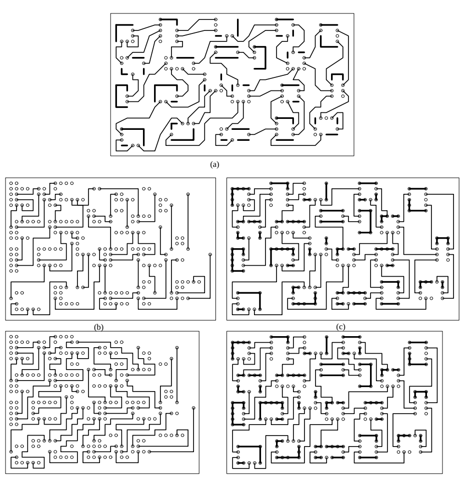

Fig. 11. Test example S_6.