Compaction on the Torus*

K. Mehlhorn, W. Rülling

08/1988

Sonderforschungsbereich 124
Fachbereich 10, Informatik
Universität des Saarlandes
D-6600 Saarbrücken
West Germany

SFB 124, B2

* **This research was supported by the DFG, grant SFB 124, TP B2**

Abstract: In this paper we introduce a general framework for compaction on a torus. This problem comes up whenever an array or row of identical cells has to be compacted. We instantiate our framework with several specific compaction algorithms: one-dimensional compaction without and with automatic jog insertion and two-dimensional compaction.

I. Introduction

A *compactor* takes as input a VLSI-Layout and produces as output an equivalent layout of smaller area. An effective compaction system frees the designer from the details of the design rules and hence increases his productivity and on the other hand produces high quality layouts. For these reasons, compaction algorithms have gained widespread attention in the VLSI-Literature ([4],[7],[9],[10],[12],[15],[17]), and are the basis for several computer-aided circuit design systems ([2],[4],[11],[19],[20]).

Regular layouts composed of rows or arrays of identical cells arise frequently in practice, e.g., bit slice architecture or systolic arrays. Let S be the cell to be replicated. We adress the following problem:

> Compact S into a cell S' such that cell S' still can be used to tile the plane (or an infinite strip)

This problem is called the compaction problem on the torus, because a layout S can be used to tile an array iff its left and right, and top and bottom boundaries are compatible, i.e. if the cell S can be drawn on a torus, cf. figure 1. The compaction problem on a torus is interesting for three reasons.

(1) Row-like and array-like arrangements of a single cell arise frequently in practice. In such an arangement it is desirable to compact all instances of the cell identically to

(2) guarantee identical electrical behavior of all instances and to

(3) allow further hierarchical processing.

Our own interest in compaction on a torus was stimulated by a Kulisch-arithmetic-chip designed by P. Lichter [14]. A central component of this design is an accumulator consisting of 1152 identical cells which are arranged into a 36 by 32 array, cf. fig. 2. The fully instantiated layout overstrained the compactor of the HILL-system (although it can handle 100 000 rectangles) and so compaction on a torus was called for. Several simple-minded approaches failed. Compacting a single cell does not guarantee tileability, compacting the layout using the algorithm for hierarchical compaction by Lengauer ([13]) does not guarantee that the instances stay identical, and compacting a single cell and insisting that the boundary stays rectangular wastes area, although it guarantees tileability. Finally, the approach of Eichenberger/Horowitz ([3]) works only for constraint based compaction without jog insertion.

In this paper we describe a framework for compaction on a torus. It can be combined with several known compaction algorithms, e.g. one-dimensional compaction ([4],[12]), one-dimensional compaction with automatic jog insertion ([15],[17]), and two-dimensional compaction ([7]), to yield specific compaction algorithms.

Our approach is very simple. Let the cell S have length L and height H. We draw S on a cylinder of circumference L and height H. If S is supposed to tile the plane (instead of a strip) then we also identify the upper and lower rim of the cylinder and obtain a torus. We now let the circumference shrink. In this way the features of the cell will move closer together until a tight cut, i.e., two features reaching their minimum separation, arises. These two features are kept at their mimimum separation from now on. We continue in this fashion until a cycle of saturated cuts around the cylinder (or torus) arises. At this point we have minimized the x-width of the cell but still guarantee that it can tile a strip (or the plane).

In section II we describe our approach in more detail and fill in some algorithmic details. We stay however on the generic level. In section III we instantiate our framework in three specific cases: one-dimension compaction without jog insertion (section III.1), one-dimensional compaction with jog insertion (section III.2) and two-dimensional compaction (section III.3).

II. Definitions and Results

We give a precise definition of the cylindrical compaction problem. A *cylindrical sketch* is a quadruple (F, W, P, L) consisting of a cylinder Z of circumference L, a finite set F of *features*, which are points (= point feature) and open straight line segments (= line feature) on the surface of Z, a finite set W of *wires*, which are simple paths on the surface of Z, and a partition P of the features F. Each block of the partition is called a *module*. Figure 3 shows an example of a sketch. When the partition P and the period L are understood we will refer to a pair (F, W) as a sketch. The features and wires of a sketch must satisfy the following conditions:

(1) Distinct features do not intersect and the endpoints of each line feature are point features.
(2) No wire may cross itself.
(3) Each wire touches exactly two features, which are point features lying at the endpoints of the wire. They are called the *terminals* of the wire.

A *point in a sketch* is a point lying on a feature. Modules form the rigid part of a layout and wires represent the flexible interconnections.

Sketches comprise the information of placement and global routing. A (detailed) *routing* of a sketch (F, W, P, L), $W = \{p_1, \dots, p_m\}$, is a sketch $(F, W', P, L), W' = \{q_1, \dots, q_m\}$, such that q_i is *homotopic* to p_i, i.e., p_i and q_i have the same endpoints and p_i can be transformed continuously into q_i without moving its endpoints and without allowing its interior to touch a feature in F, and such that the q_i's satisfy the constraints of the particular wiring model used. We consider only the grid model in this paper; our results extend however to any polygonal wiring norm. In the *grid model* wires are rectilinear paths with a minimum vertical and horizontal separation of 1.

A *cut* C is any open line segment connecting two points of the sketch, say p and q, and not intersecting any feature. The *density* of cut C is the number of crossings of C by wires which are enforced by the topology of the sketch, cf. figure 4. Crossings of C which can be removed by deforming the wires do not contribute to the density. The *capacity* of a cut in the grid model is given by $\max\{x\text{-length}(C), y\text{-length}(C)\} - 1$, where x-length(C) (y-length(C)) is the length of C in horizontal (vertical) direction, cf. figure 5. A cut is called *safe* if its density does not exceed its capacity and it is called *tight* or *saturated* if its density is equal to its capacity. The following theorem was proved by Cole/Siegel ([1]) and Leiserson/Maley ([8]) for the grid model.

Theorem 1. *A sketch has a routing iff all cuts of the sketch are safe.*

Actually, the results are slightly stronger. Let us call a cut $\overline{pq}$ *critical*, if either p and q are point features or at most one of them lies on a line feature and the line segment $\overline{pq}$ is perpendicular to that line feature. Then a sketch is routable iff all critical cuts are safe.

With every cylindrical sketch S we can associate an infinite planar sketch S^∞ as follows. Let R be any vertical line on the cylinder. Then S^∞ is obtained by unrolling the cylinder and then tiling a strip with the unrolled cylinder, cf. figure 5.

We are now ready to define the one-dimensional cylindrical compaction problem. The goal of compaction is to displace the modules in x-direction such that the resulting sketch is routable and has minimal period. Let $S = (F, W, P, L)$ be a *routable cylindrical sketch* and let S^∞ be the associated planar sketch. We denote the different instances of a feature f by $f_i, i \in \mathbb{Z}$. A displacement (or configuration) of S^∞ is given by a vector $d \in \mathbb{R}^{F\times\mathbb{Z}}$; $d(f_i)$ is the displacement of feature f_i. Of course, not all displacements make sense. Firstly, features in the same module must be displaced by the same amount and therefore we must have $d(f_i) = d(g_i)$ for any two feature instances in the same module. Secondly, features should not cross over during compaction and we therefore must have $x_p + d(f_i) < x_q + d(g_j)$ for any two points $p = (x_p, y_p)$ and $q = (x_q, y_q)$ where $x_p < x_q$ and $y_p = y_q$ and p lies on a feature f_i and q lies on a feature g_j. Let d be a configuration satisfying the two constraints above. We can now define the sketch $S^\infty(d)$ in a natural way. A point p on feature instance f_i with coordinates (x_p, y_p) in the sketch S^∞ has coordinates $(x_p + d(f_i), y_p)$ in $S^\infty(d)$ and the wires in $S^\infty(d)$ have the "same" homotopies as in S^∞; cf. [15] for a more precise definition. The *configuration space* $C(S) \subseteq \mathbb{R}^F \times \mathbb{R}$ of a cylindrical sketch S consists of all pairs $(d, \delta), d \in \mathbb{R}^F, \delta \in \mathbb{R}$, such that the configuration $\tilde{d} \in \mathbb{R}^{F\times\mathbb{Z}}$ with $\tilde{d}(f_i) = d(f) + i\delta, f \in F$ and $i \in \mathbb{Z}$, satisfies the two constraints above and $S^\infty(\tilde{d})$ is routable. Note that the pair $(0, L)$, where 0 is the zero-vector, belongs to $C(S)$, since the sketch S is assumed to be routable. Also note that the sketch $S^\infty(\tilde{d})$ can be wrapped around a cylinder of circumference δ and hence gives rise to a cylindrical sketch which we denote $S(d, \delta)$. The *essential configuration* space $C_0(S)$ of a sketch S consists of that connected component of $C(S)$ which contains the pair $(0, L)$, i.e., a configuration (d, δ) belongs to $C_0(S)$ if the cylindrical sketch $S(d, \delta)$ can be obtained from $S = S(0, L)$ by continuously shrinking the cylinder and deforming the layout drawn on its surface whilst maintaining the routability of the sketch.

Definition: One-dimensional cylindrical compaction problem

> **Input:** A routable cylindrical sketch $S = (F, W, P, L)$
>
> **Output:** A configuration $(d, \delta) \in C_0(S)$ such that δ is minimal.

Theorem 2. *Let $S = (F, W, P, L)$ be a routable cylindrical sketch. Then the essential configuration space $C_0(S)$ of the sketch S is a convex polyhedron.*

Proof: In [15] the analogous result for planar sketches was proved. Because of the correspondence between cylindrical sketches and periodic planar sketches described above the result carries over. ∎

We now state our main results. Let $m = |F|$ be the number of features.

Theorem 3 (Cylindrical Compaction with Automatic Jog Insertion) .
In the grid model the cylindrical compaction problem can be solved in time

$$O(m^3 W_{max}^2 \log m + K \log m) = O(m^4 W_{max}^2 \log m)$$

where $W_{max} = 1 + \lfloor H/\Delta_{min} \rfloor$, H is the height of the sketch S, Δ_{min} is the period of the compacted layout and K is the number of times a feature moves across a critical cut during compaction. ∎

The quantity K is a measure of how much the sketch changes during compaction. We believe that the bound $K \leq m^4 W_{max}^2$ which we derive in section IV is overly pessimistic.

Compaction without jog insertion is a special case of theorem 3. Let us assume that wires are specified as rectilinear polygonal paths; view vertical wire pieces as modules and horizontal wire pieces as wires in the sense of the definition of a sketch, cf. figure 6. Then the compaction of such a sketch is tantamount to compaction without jog insertion.

Theorem 4.
Cylindrical compaction without jog insertion can be solved in time $O(m^2 \log m)$.

Cylindrical compaction without jog insertion was previously considered by Eichenberger/ Horowitz [3]. They did not analyse their algorithm.

Our approach can also handle maximum and minimum distance constraints which are specified by the user as long as the constraints are satisfied by the initial layout S. Since *toroidal compaction* amounts to cylindrical compaction in the presence of equality constraints between the upper and the lower cell boundary our algorithms carry over to toroidal compaction with unchanged running time. Finally, we want to mention that the algorithm underlying theorem 4 can be used for Kedem/Watanabe-like two-dimensional compaction ([7]).

III. Compaction on a Torus: the Framework

In this section we describe the general framework for compaction on a torus. For simplicity, we deal only with the cylinder. Let $S = (F, W, P, L)$ be the cylindrical sketch to be compacted, cf. fig. 3; let $\Delta = L$.

The central concept of our approach is shrinking which we define next. Let $S(\Delta)$ be the sketch obtained for the circumference Δ of the cylinder and let R be a vertical line on the cylinder which we call the reference line. For a feature f let $p(f, \Delta)$ be the distance from f to the reference line when going to the left starting in f. The local meaning of "left" and "right" is defined by viewing the cylinder from the outside. We refer to $p(f, \Delta)$ as the position of f in the sketch $S(\Delta)$. For a cut C let the wrapping number $w(C, \Delta)$ be the number of intersections between C and the reference line R. We extend the concept of wrapping number to features as follows. Consider an auxiliary digraph G_A with vertex set $F \cup \{R\}$. For every feature f there is an edge (R, f) of cost 0 and for every saturated cut C with endpoints f and g, where the left-to-right orientation is from f to g, there is an edge (f, g) of cost $w(C, \Delta)$. We denote such a cut C by $\overline{fg}$; note that this notation is ambiguous since only the endpoints together with the wrapping number identify a cut. Let us assume for the moment that the auxiliary graph G_A is acyclic; the other case is treated in the proof of lemma 3 below. Let $T(\Delta)$ be a longest path tree with root R in the auxiliary graph G_A, let $w(f, \Delta)$ be the length of a longest path from R to f in G_A and let $d(f, \Delta) = \Delta \cdot w(f, \Delta) + p(f, \Delta)$, cf. figure 7 for an illustration. We refer to $w(f, \Delta)$ as the wrapping number of f and to $d(f, \Delta)$ as the distance from R to f in $S(\Delta)$. With these concepts it is now easy to define the sketch $S(\Delta - \epsilon)$. The position $p(f, \Delta - \epsilon)$ of feature f in $S(\Delta - \epsilon)$ is given by $d(f, \Delta) \bmod (\Delta - \epsilon)$. The wire homotopy in $S(\Delta - \epsilon)$ is defined in the natural way by considering the continuous transformation (ϵ grows starting at 0) from the positions $p(f, \Delta)$ to the positions $p(f, \Delta - \epsilon), f \in F$.

Shrinking the circumference of the cylinder can be visualized as follows. We cut the cylinder at the reference line R and obtain a single copy of the sketch with left boundary R and right boundary R'. We then move each feature f to the right by an amount $\epsilon \cdot w(f,\delta)$ and move R' to the left by ϵ, cf. fig. 8. For small ϵ no feature will cross R' and hence a cylindrical layout is obtained by identifying R and R'.

Lemma 1. *a) Let $C = \overline{fg}$ be a cut and let $x(C,\Delta) := p(g,\Delta) - p(f,\Delta) + w(C,\Delta)\cdot\Delta$ be the x-length of C in $S(\Delta)$. Then the x-length of C in $S(\Delta-\epsilon)$ is given by $x(C,\Delta-\epsilon) = x(C,\Delta) + \epsilon(w(g,\Delta) - w(f,\Delta) - w(C,\Delta))$.*

b) If $T(\Delta)$ exists then $S(\Delta-\epsilon)$ is legal for $\epsilon > 0$ sufficiently small.

Proof: a) Let $\Delta' = \Delta - \epsilon$. Then

$$\begin{aligned} x(C,\Delta') &= p(g,\Delta') - p(f,\Delta') + w(C,\Delta')\cdot\Delta' \\ &= (p(g,\Delta) + w(g,\Delta)\cdot\Delta) \bmod \Delta' - (p(f,\Delta) + w(f,\Delta)\cdot\Delta) \bmod \Delta' \\ &\quad + w(C,\Delta')\cdot\Delta' \\ &= (p(g,\Delta) + w(g,\Delta)\cdot\epsilon) \bmod \Delta' - (p(f,\Delta) + w(f,\Delta)\cdot\epsilon) \bmod \Delta' \\ &\quad + w(C,\Delta')\cdot\Delta - w(C,\Delta')\cdot\epsilon \end{aligned}$$

Let us assume for simplicity that $p(g,\Delta) + w(g,\Delta)\cdot\epsilon < \Delta'$ and $p(f,\Delta) + w(f,\Delta)\cdot\epsilon < \Delta'$; the other case is similar and left to the reader. Then $w(C,\Delta') = w(C,\Delta)$ and hence $x(C,\Delta') = x(C,\Delta) + \epsilon(w(g,\Delta) - w(f,\Delta) - w(C,\Delta))$.

b) Let $C = \overline{fg}$ be any cut. Clearly, if C is not tight in $S(\Delta)$ then C is not tight in $S(\Delta-\epsilon)$ for ϵ sufficiently small. If C is tight in $S(\Delta)$ then there is an edge (f,g) of cost $w(C,\Delta)$ in the auxiliary graph and hence $w(g,\Delta) \geq w(f,\Delta) + w(C,\Delta)$ by the definition of wrapping number. Thus $x(C,\Delta-\epsilon) \geq x(C,\Delta)$. Since the y-coordinates of the features do not change during shrinking the capacity of C does not go down when passing from $S(\Delta)$ to $S(\Delta-\epsilon)$. Also, the density of C does not change for ϵ sufficiently small. Thus $S(\Delta-\epsilon)$ is legal for ϵ sufficiently small. ∎

Lemma 1 is the basis for our compaction algorithm. If $T(\Delta)$ exists, then the shrinking process yields a legal sketch of smaller period. This leads to the following algorithm.

$\Delta \leftarrow w$; $S(\Delta) \leftarrow S$ (* the initial sketch S has period w *)
while $T(\Delta)$ exists
do let $\epsilon > 0$ be maximal such that $S(\Delta-\epsilon)$ is legal;
  compute $S(\Delta-\epsilon)$ and $T(\Delta-\epsilon)$;
  $\Delta \leftarrow \Delta - \epsilon$;
od;

It remains to prove termination (lemma 2) and correctness (lemma 3).
Let $W_{max}(\Delta) = \max\{w(C,\Delta); C = \overline{fg}$ is a tight cut and there is no sequence $f_0,\ldots,f_k$ such that $f = f_0, g = f_k, C_i = \overline{f_i f_{i+1}}$ is tight for $0 \leq i < k$, and $w(C,\Delta) = \sum_i w(C_i,\Delta)\}$, i.e., $W_{max}(\Delta)$ is the maximal wrapping number of any cut which cannot be replaced by a sequence of tight cuts of smaller wrapping number. We prove upper bounds for W_{max} in various compaction models in section IV. In particular, $W_{max}(\Delta) = 1$ for compaction without jog insertion and $W_{max}(\Delta) \leq 1 + \lfloor H/\Delta \rfloor$ for compaction with jog insertion in the grid model.

Lemma 2. *a)* $0 \le w(f,\Delta) \le m \cdot W_{max}(\Delta)$ *for all* f *and* Δ *and* $w(f,\Delta)$ *is non-decreasing for every* f.

b) The number of iterations is bounded by $m^2 W_{max}(\Delta_{min})$ *where* Δ_{min} *is the period of the final sketch.*

Proof: a) The bounds $0 \le w(f,\Delta)$ and $w(f,\Delta) \le m \cdot W_{max}(\Delta)$ follow immediately from the definition of wrapping numbers. We show next that $w(f,\Delta)$ is non-decreasing for every f. Consider any $S(\Delta)$ and let ϵ be maximal such that $S(\Delta - \epsilon)$ is legal. Then there must be a cut $C = \overline{fg}$ which is tight in $S(\Delta - \epsilon)$ and oversaturated in $S(\Delta - \epsilon - \delta)$ for $\delta > 0$. Thus $w(g,\Delta) - w(f,\Delta) - w(C,\Delta) < 0$ by lemma 1a.

Consider the layouts $S(\Delta - \delta)$ where $0 \le \delta < \epsilon$. In these layouts exactly the cuts $D = \overline{hk}$ with $w(k,\Delta) = w(h,\Delta) + w(D,\Delta)$ are tight. This follows from lemma 1a and the definition of ϵ. In particular, all cuts in $T(\Delta)$ stay tight. Moreover, the tree $T(\Delta - \delta), 0 \le \delta < \epsilon$, is independent of δ. This can be seen as follows. As we increase δ from 0 to ϵ, the wrapping number of a feature h increases by one whenever h moves across the reference line R on the cylinder. Note that in this case the wrapping number of all cuts incident to h and leaving h to the left goes up by one and the wrapping number of the cuts leaving h to the right goes down by one. Thus the longest path tree does not change. The argument also shows that the quantity $w(g,\Delta - \delta) - w(f,\Delta - \delta) - w(C,\Delta - \delta)$ is a constant independent of δ.

In $S(\Delta - \epsilon)$ the cut C becomes tight and is added to the auxiliary graph. Since $w(g,\Delta - \delta) - w(f,\Delta - \delta) - w(C,\Delta - \delta) = w(g,\Delta) - w(f,\Delta) - w(C,\Delta) < 0$ and hence $w(f,\Delta - \delta) + w(C,\Delta - \delta) > w(g,\Delta - \delta)$ for all $\delta, 0 \le \delta < \epsilon$, there is now a longer path to g in the auxiliary graph. Thus we obtain $T(\Delta - \epsilon)$ by replacing in $T(\Delta)$ the edge (= cut) currently ending in g by an edge corresponding to C. Also, $w(g,\Delta - \epsilon) = w(f,\Delta - \epsilon) + w(C,\Delta - \epsilon)$.

b) We have shown in part a) that the values $w(f,\Delta)$ are non-decreasing, that at least one such value is increased in each iteration and that $0 \le w(f,\Delta) \le m \cdot W_{max}(\Delta_{min})$. Thus the number of iterations is bounded by $m^2 W_{max}(\Delta_{min})$. ∎

Lemma 3. *The algorithm constructs a sketch of minimum period.*

Proof: Let S_{fin} be the final sketch. It is clearly reachable (the algorithm shows how) from the initial sketch S by a continuous transformation which only passes through legal configurations. Thus $S_{fin} = S(d_0, \Delta_0)$ for some $(d_0, \Delta_0) \in C_0(S)$. In S_{fin} there must be a sequence $f_0, \dots, f_k$ of features such that $f_0 = f_k$, and the cuts $C_i = \overline{f_i f_{i+1}}, 0 \le i < k$, are tight. Let x_i be the x-length of C_i and let w_i be its wrapping number. Then $x_0 + \ldots + x_{k-1} = \Delta_0(w_0 + \ldots + w_{k-1})$. Let $S' = S(d,\delta)$ with $(d,\delta) \in C_o(S)$ be arbitrary. For $0 \le \lambda \le 1$ consider the configuration $(d(\lambda), \delta(\lambda))$ with $d(\lambda) = (1-\lambda)d_0 + \lambda d$ and $\delta(\lambda) = (1-\lambda)\Delta_0 + \lambda\delta$. Then $(d(\lambda), \delta(\lambda)) \in C_0(S)$ since $C_0(S)$ is convex by theorem 1. Next observe that the cuts C_i exist in $S(d(\lambda), \delta(\lambda))$ for λ sufficiently small and that their density is the same as in S_{fin}. Thus their capacity must be no smaller than in S_{fin} and hence their x-length must be no smaller than in S_{fin}. Their total x-length is $\delta(\lambda)(w_0 + \ldots + w_{k-1})$ and hence $\delta(\lambda) \ge \Delta_0$ for λ small. Thus $\delta \ge \Delta_0$ and S_{fin} has minimal period. ∎

At this point we have proved termination and correctness of our generic compaction algorithm. We fill in some more algorithmic detail next. The data structures are the longest path tree $T = T(\Delta)$ and the set $A = A(\Delta)$ of cuts. With every cut $C \in A$ we associate the minimal value $\epsilon(C)$ of ϵ such that C becomes tight in $S(\Delta - \epsilon)$. The value $\epsilon(C)$ is easily computed from the density and the capacity of C in $S(\Delta)$ using lemma 1a. Let $\epsilon_1 = min\{\epsilon(C); C \in A\}$.

For every feature f let $\epsilon(f)$ be the minimal value of ϵ such that $p(f, \Delta - \epsilon) \bmod (\Delta - \epsilon) = 0$, i.e. f moves across the reference line in $S(\Delta - \epsilon)$. Let $\epsilon_2 = \min\{\epsilon(f); f \in F\}$. Finally, let $\epsilon_3 = \min\{\epsilon; A(\Delta - \epsilon) \neq A(\Delta)\}$ and let $\epsilon = min(\epsilon_1, \epsilon_2, \epsilon_3)$. We distinguish three cases according to whether $\epsilon = \epsilon_1, \epsilon = \epsilon_2$ or $\epsilon = \epsilon_3$. The three cases are not mutually exclusive.

Case 1, $\epsilon = \epsilon_1$ **:** Let $\epsilon = \epsilon(C)$ and $C = \overline{fg}$. Let T_g be the subtree of T rooted at g. We perform the following actions:

1) If $f \in T_g$ then **STOP**, $T(\Delta - \epsilon)$ does not exist.
2) Increase $w(h)$ by $w(f) + w(C) - w(g)$ for every feature $h \in T_g$; here $w(h)$ denotes the current wrapping number of feature $h \in T_g$.
3) Delete the current edge ending in g from T and add the edge (f, g).
4) Recompute $\epsilon(D)$ for every cut D incident to a feature $h \in T_g$, recompute $\epsilon(h)$ for every $h \in T_g$ and recompute ϵ_1, ϵ_2 and ϵ_3.

Case 2, $\epsilon = \epsilon_2$ **:** Let $\epsilon = \epsilon(f), f \in F$.

5) Increase $w(f)$ and $w(C)$ by one for all cuts C leaving f to the left and decrease $w(C)$ by one for all cut leaving f to the right. Update $\epsilon(f)$ and ϵ_2.

Case 3, $\epsilon = \epsilon_3$ **:** A either grows or shrinks at $\Delta - \epsilon$.

Case 3.1. A shrinks, say cut C disappears, cf. figure 9.

6) Delete C from A. Update ϵ_1 and ϵ_2.

Case 3.2. A grows, say cut C appears, cf. figure 9.

7) Add C to A, compute $\epsilon(C)$ and update ϵ_1 and ϵ_3.

Remark: In the high level description of the algorithm cases 2 and 3 did not appear because case 2 does not change the longest path tree and the values $\epsilon(C)$. Case 3.1 either removes an unsaturated cut or occurs together with case 1. Case 3.2 creates only unsaturated cuts, cf. figure 9. ∎

IV. Specific Compaction Algorithms

In this section we derive specific compation algorithms from the general framework of section III.

IV.1. One-dimensional Compaction without Jog Insertion

We assume that wires are specified as rectilinear polygonal paths in the input sketch S. We treat vertical wire segments as modules and horizontal wire segments as wires in the sense of the definition of sketch. The only cuts which have to be considered are horizontal cuts connecting pairs of features which are visible from each other. Thus there are only $O(m)$ critical cuts, the set A of cuts does not change and $W_{max} = 1$ since no cut can wrap around twice.

For a feature h let $deg(h)$ be the number of cuts incident to h. Then $\sum_{h \in F} deg(h) = O(m)$ since the set of cuts defines a planar graph on the set of features. Also, actions 4 and 5 are executed at most m times for each h by lemma 2a and hence the total cost is $\sum_{h \in F} m\ deg(h) \cdot \log m = O(m^2 \log m)$; the $\log m$ factor results from the fact that a change of $\epsilon(D)$ requires a heap operation. This proves theorem 4.

IV.2. One-dimensional Compaction with Jog Insertion

In order to apply our generic algorithm we need a bound on W_{max} and we need a way to manage the set A of cuts.

Lemma 4. *Let S be a cylindrical sketch of height H and period Δ. Let W_{max} be the maximal wrapping number of any tight cut which cannot be replaced by a sequence of shorter tight cuts. Then $W_{max} \leq 1 + \lfloor H/\Delta \rfloor$ in the grid model.*

Proof: We use the concept of shadowing, cf. ([1],[16]). Let $C = \overline{fg}$ be a saturated cut with $w(C, \Delta) > 1 + H/\Delta$, cf. figure 10. Consider the straight line segments $C' = \overline{fg'}$ and $C'' = \overline{g'g}$ with $w(C', \Delta) = w(C, \Delta) - 1$ and $w(C'', \Delta) = 1$. The triangle with sides C, C' and C'' must contain a feature h ($h = g'$ is possible) such that the line segments $D' = \overline{fh}$ and $D'' = \overline{hg}$ with $w(D', \Delta) + w(D'', \Delta) = w(C, \Delta)$ are both cuts and the triangle with sides C, D' and D'' contains no features. Then the capacity of C is equal to the sum of the capacities of D' and D'' plus 1, since D' and D'' have both slope at most one, and the density of C is at most the sum of the densities of D' and D'' plus 1. Thus D' and D'' are tight and can replace C. We conclude $W_{max} \leq 1 + H/\Delta$. ∎

We store the set of cuts as in [17], i.e., for each point feature f we store the set of critical cuts incident to f in clockwise order in a balanced tree. The results of [17] imply that actions 1 to 5 take time $O(deg(h) \log m)$ for each feature h whose wrapping number is increased where $deg(h)$ is the number of cuts incident to h and that actions 6 and 7 take time $O(\log m)$ each. Let K be the number of times actions 6 and 7 are performed. Then the running time is $O(K \log m + m \cdot (mW_{max}) \cdot (mW_{max} \log m))$ since the maximal wrapping number of any feature is $m \cdot W_{max}$ by lemma 2 and since the maximal number of critical cuts incident to any feature is mW_{max}; note that for each feature g there can be up to W_{max} different cuts with endpoints f and g. Finally, note that $K \leq 3m^4 W_{max}^2$. This can be seen as follows. Consider a pair (f, g) of features and one of the W_{max} possible cuts C with endpoints f and g. Any feature h can cross C at most $3mW_{max}$ times since between consecutive crossings the wrapping number of at least one of the three features must have been increased. Thus the running time is $O(K \log m + m^3 W_{max}^2 \log m) = O(m^4 W_{max}^2 \log m)$ and theorem 3 is proved. We believe that our bound on K is overly pessimistic.

IV.3. Two-dimensional Compaction without Jog Insertion

Finally, we describe how two-dimensional compaction can be done on the torus. Kedem/Watanabe ([7]) describe a branch-and-bound approach to the two-dimensional compaction problem in the plane. That algorithm uses one-dimensional compaction to compute the lower bounds in the bound step. This readily extends to toroidal compaction by virtue of theorem 3. The difficulty of two-dimensional compaction is the interaction between horizontal and vertical movement of features. For example, one has to decide whether it is better to separate two features f and g by a horizontal or vertical cut, cf. fig. 11. Therefore, in the compaction method presented in [7] decision variables $d_{f,g} \in \{0, 1\}$ are introduced for every pair of interacting features. The value of $d_{f,g}$ determines whether a horizontal or vertical separation between f and g has to be used. Let d denote the vector of all decision variables. Then for every fixed d the two-dimensional compaction problem can be solved by seperately applying one-dimensional compaction in horizontal and vertical direction. This works on the torus as well as in the plane. The remaining problem of computing the optimal decision vector d can be solved by the branch-and-bound algorithm. By this method one starts with all decision variables being undefined, and then tries to fix the variables in such a way that the layout area produced by the two passes of one-dimensional compaction will be minimal. Essentially this is done by investigating a decision tree and prunning away subtrees that cannot improve the objective function. For this purpose we need lower bounds and upper bounds of the layout area. To obtain a lower bound we compute the minimal horizontal period $\Delta_{min}^{(hor)}$ and the minimal vertical period $\Delta_{min}^{(vert)}$ by one-dimensional cylindrical compaction of the given sketch ignoring cuts $\overline{fg}$ corresponding to decision variables that are not yet fixed. Obviously, the product $\Delta_{min}^{(hor)} \cdot \Delta_{min}^{(vert)}$ is a lower bound of the compacted layout area. Upper bounds can be obtained by inserting a horizontal or vertical cut for every unfixed decision variable. This way the branch-and-bound method can be used for two-dimensional compaction with and without jog insertion.

Acknowledgment: We thank M. Jerrum, St. Näher and Th. Schilz for many helpful discussions. M. Jerrum suggested that toroidal compaction is the adequate approach to compacting regular structures.

V. References

[1] Cole, R., and Siegel, A., "River routing every which way, but loose", Proceedings of the 25th Annual Symposium on Foundations of Computer Science, October 1984, pp. 65–73

[2] Dunlop, A.E., "SLIP: symbolic layout of integrated circuits with compaction", Computer Aided Design, Vol. 10, No. 6, 1978, pp. 387–391

[3] Eichenberger, P., and Horowitz, M., "Toroidal Compaction of Symbolic Layouts for Regular Structures", 1987 IEEE, ICCAD, pp. 142–145

[4] Hsueh, M.Y., "Symbolic Layout and Compaction of Integrated Circuits", Ph.D. thesis, EECS Division, University of California, Berkeley, 1979

[5] Gao, S., Jerrum, M., Kaufmann, M., Mehlhorn, K., Rülling, W., Storb, C., "On homotopic river routing", BFC 87, Bonn, 1987

[6] Kaufmann, M., and Mehlhorn, K., "Local Routing of Two-Terminal Nets", 4th STACS 87, LNCS 247, pp. 40–52

[7] Kedem, G., and Watanabe, H., "Optimization techniques for IC layout and compaction", Technical Report 117, Computer Science Department, University of Rochester, 1982

[8] Leiserson, C.E., and Maley, F.M., "Algorithms for routing and testing routability of planar VLSI layouts", Proceedings of the 17th Annual ACM Symposium on Theory of Computing, 1985, pp. 69–78

[9] Leiserson, C.E., and Pinter, R.Y., "Optimal placement for river routing", SIAM Journal on Computing, Vol. 12, No. 3, 1983, pp. 447–462

[10] Lengauer, T., "Efficient algorithms for the constraint generation for integrated circuit layout compaction", Proceedings of the 9th Workshop on Graphtheoretic Concepts in Computer Science, 1983

[11] Lengauer, T., and Mehlhorn, K., "The HILL system: a design environment for the hierarchical specification, compaction, and simulation for integrated circuit layouts", Proceedings, Conference on Advanced Research in VLSI, 1984

[12] Lengauer, T., "On the solution of inequality systems relevant to IC layout", Journal of Algorithms, Vol. 5, No. 3, 1984, pp. 408–421

[13] Lengauer, T., "The Complexity of Compacting Hierarchically Specified Layouts of Integrated Circuits", 1982 FOCS, pp. 358–368

[14] Lichter, P., "Ein Schaltkreis für die Kulischarithmetik", Diplomarbeit, Universität des Saarlandes, 1988, a summary appears in GI-Jahrestagung 1988

[15] Maley, F.M., "Compaction with Automatic Jog Insertion", 1985 Chapel Hill Conference on VLSI

[16] Maley, F.M., "Single-Layer Wire Routing", Ph.D. thesis, MIT, 1987

[17] Mehlhorn, K., and Näher, St., "A Faster Compaction Algorithm with Automatic Jog Insertion", Proceedings of the 1988 MIT VLSI Conference

[18] Pinter, R.Y., "The Impact of Layer Assignment Methods on Layout Algorithms for Integrated Circuits", Ph.D. thesis, MIT Department of Electrical Engineering and Computer Science, 1982

[19] Rülling, W., "Einführung in die Chip-Entwurfssprache HILL", Techn. Bericht 04/1987, SFB124, Univ. des Saarlandes, 1987

[20] Williams, J.D., "STICKS — a graphical compiler for high level LSI design", National Computer Conference, 1978, pp. 289–295

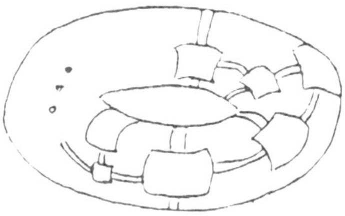

Figure 1 :
A toroidal sketch.

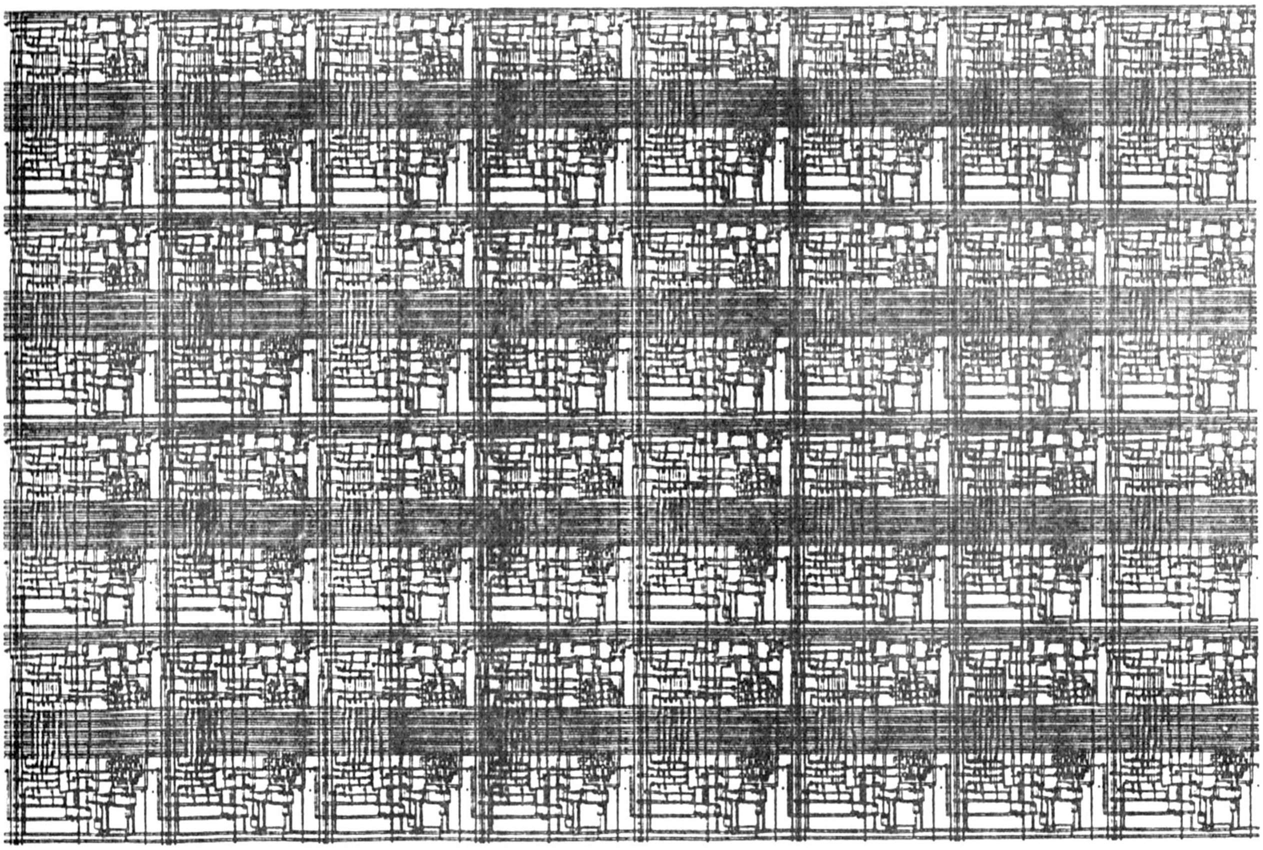

Figure 2 :
Part of a large accumulator consisting of identical cells. This layout has been computed by compaction on the torus without jog insertion.

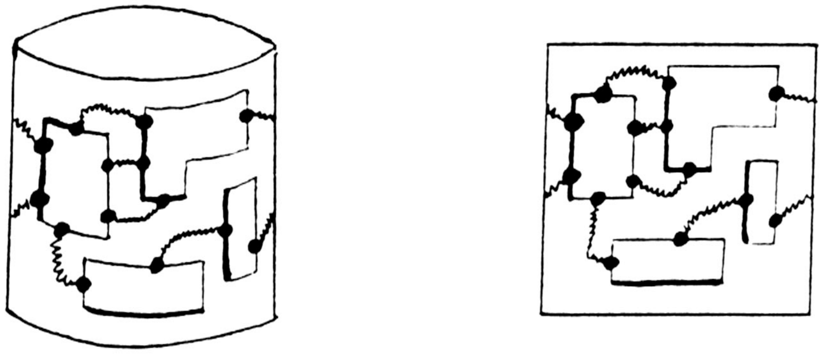

Figure 3 :
A typical cylindrical sketch (F, W, P, L) and its representation in the plane. Dark points and line segments are features, light lines are conceptual module boundaries. Formally, the connected components formed by dark and light lines are the blocks of the partition P. Wires are shown as wiggled lines.

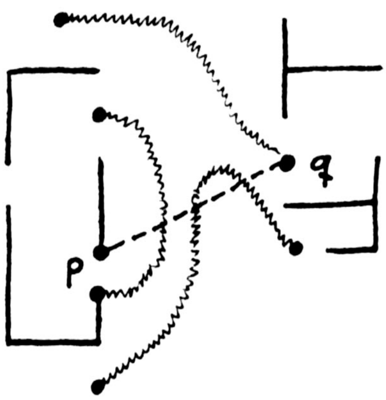

Figure 4 :
A portion of a sketch with cut $\overline{pq}$. The flow across $\overline{pq}$ is 1.

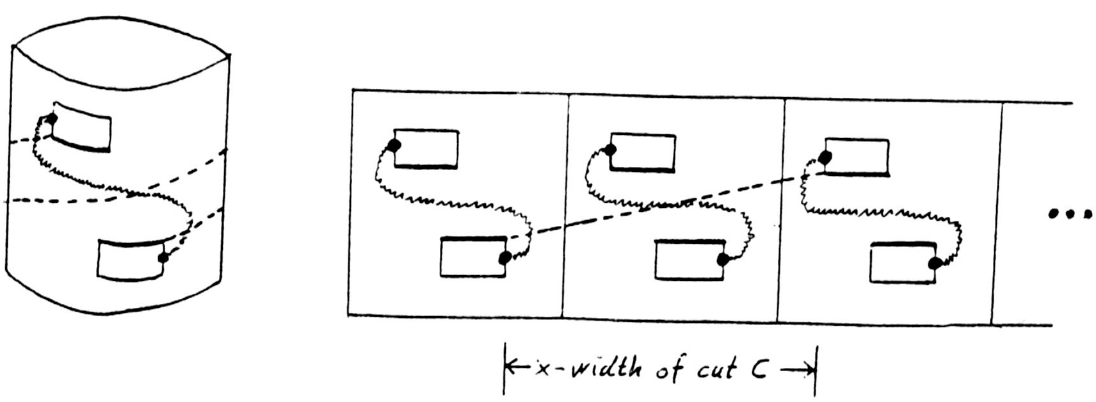

Figure 5 :
The unrolled version S^∞ of a cylindrical sketch S. The shown cut C between f and g has density 3.

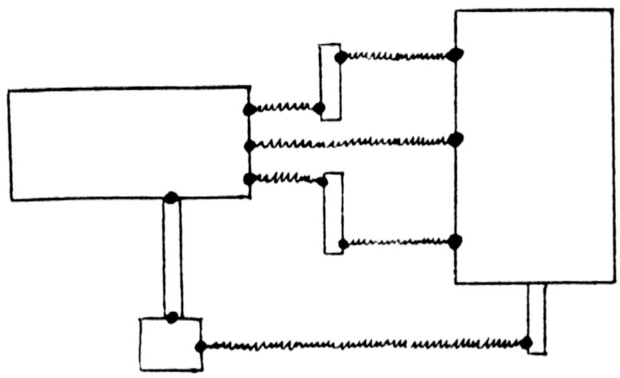

Figure 6 :
A sketch for compaction without jog insertion. Vertical wire segments are treated as modules.

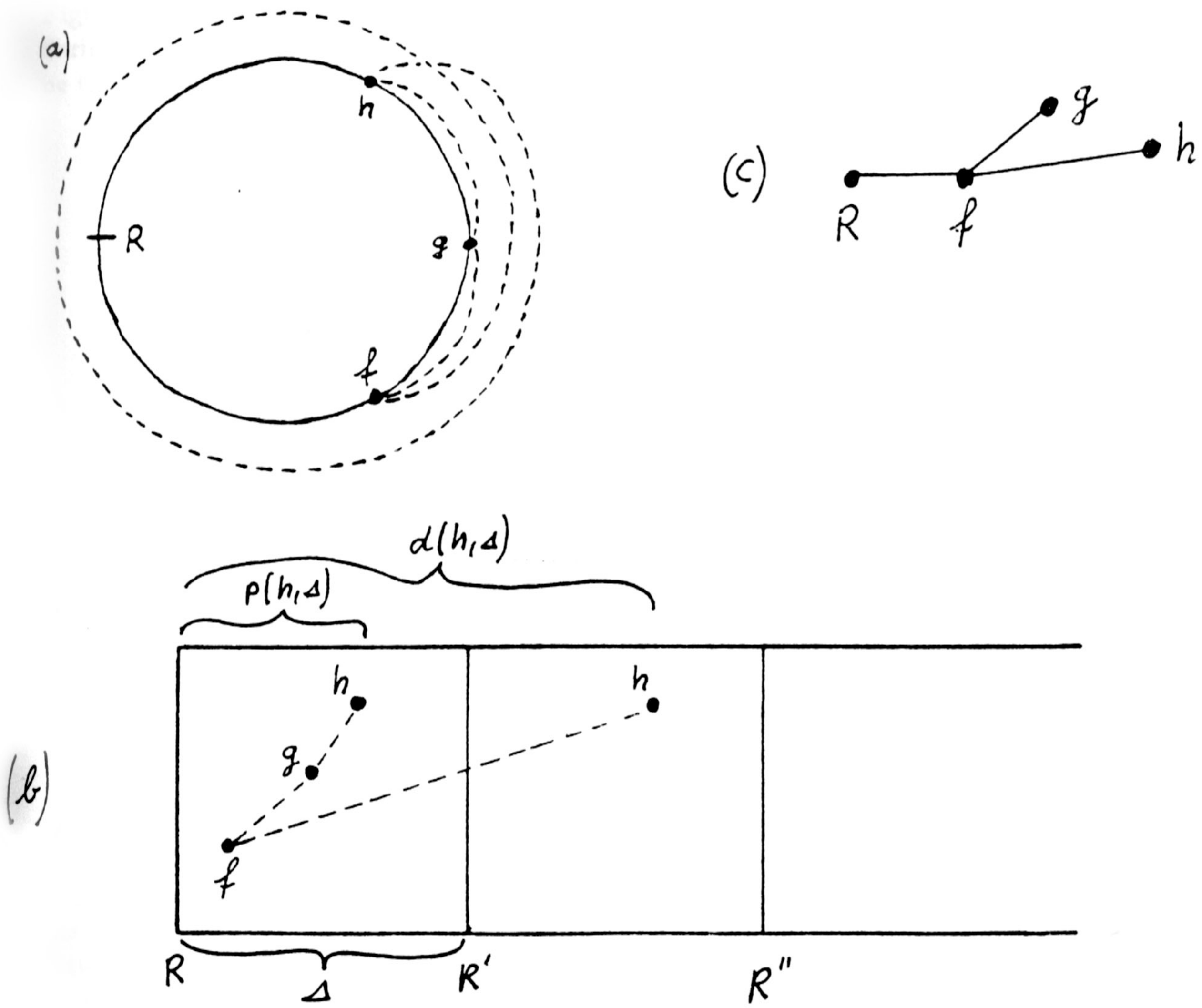

Figure 7 :
Part a) shows a top view of the cylinder. There are saturated cuts $\overline{fg}, \overline{gh}$ and $\overline{fh}$. The latter cut has wrapping number 1 and the former two have wrapping number 0. Part b) shows the unrolled picture. Part c) shows the tree $T(\Delta)$.

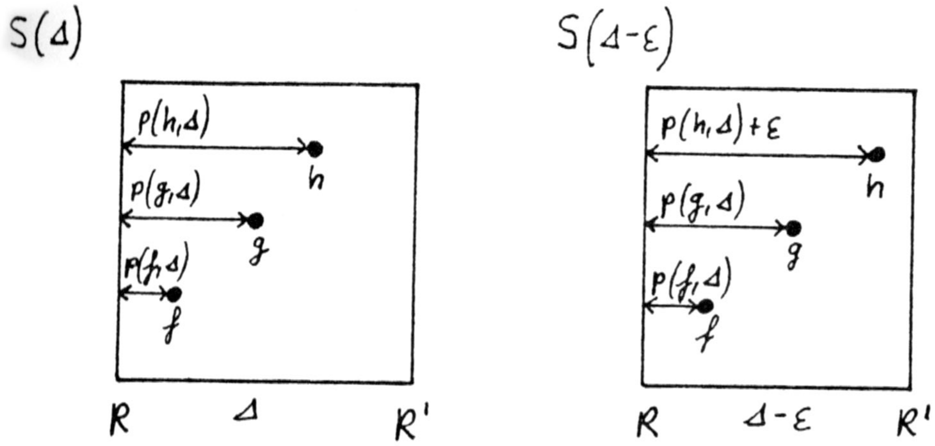

Figure 8 :
Shrinking the circumference Δ by ϵ in the example of figure 7. Note that the cut $\overline{gh}$ may be tight in $S(\Delta)$ but it is not tight in $S(\Delta - \epsilon)$.

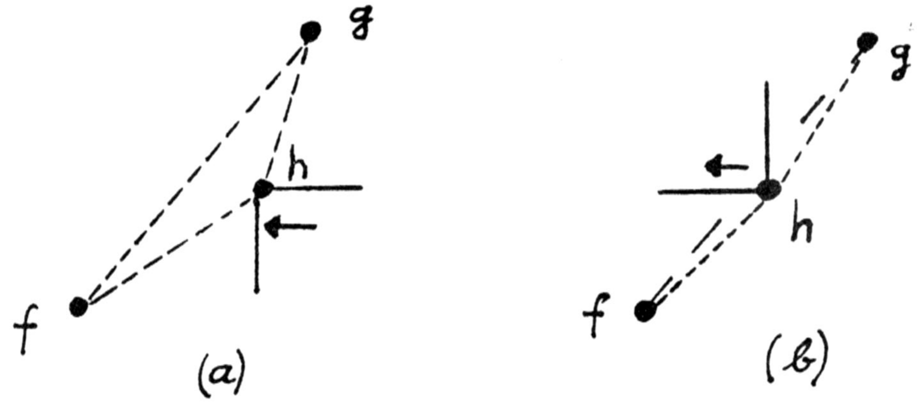

Figure 9 :
In (a) h moves to the left relative to the cut $\overline{fg}$ and hits this cut in $S(\Delta - \epsilon)$. The cut $\overline{fg}$ will then disappear. If the cut $\overline{fg}$ is tight then both cuts $\overline{fh}$ and $\overline{fg}$ will be tight in $S(\Delta - \epsilon)$ and hence cases 1 and 3.1. arise together. In b) h also moves to the left with respect to the line segment $\overline{fg}$. Thus the cut $\overline{fg}$ will arise, say in $S(\Delta - \epsilon)$. The cut $\overline{fg}$ cannot be saturated because o.w. the cuts $\overline{fh}$ and $\overline{hg}$ would be saturated in $S(\Delta - \epsilon)$ and hence $\overline{fh}$ would be oversaturated in $S(\Delta - \epsilon - \delta)$ for $\delta > 0$ small. Thus $(f, h) \in T(\Delta - \epsilon)$ and g would never become visible from f.

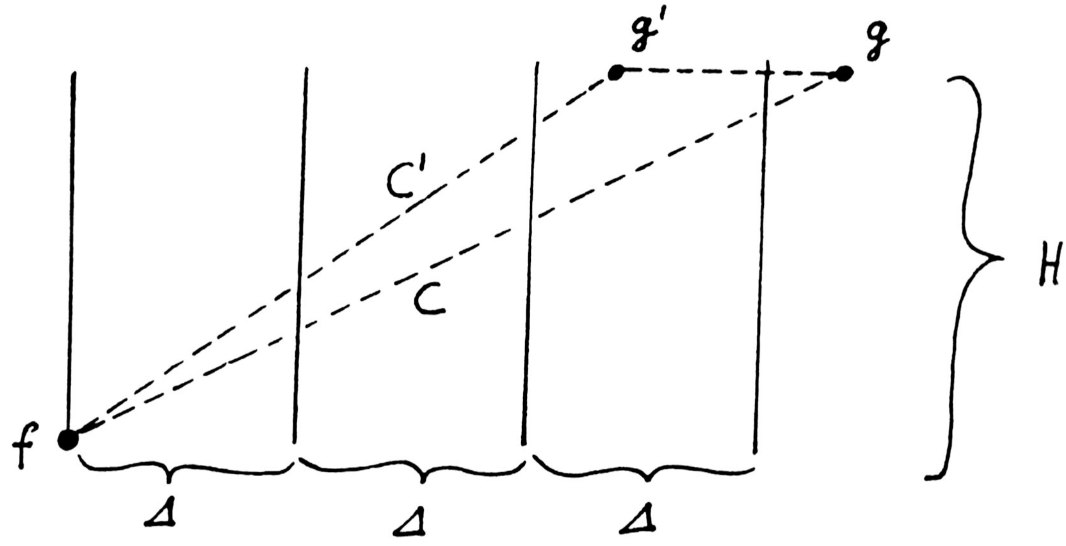

Figure 10 :
The cut $C = \overline{fg}$ has wrapping number $\geq 1 + H/\Delta$. g' is equal to g and the line segment C' has wrapping number one less than C.

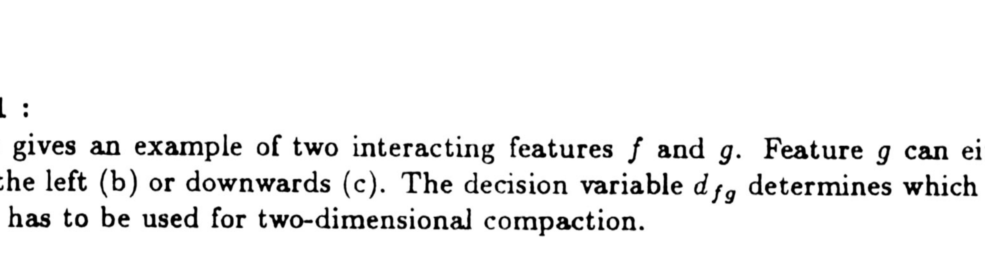

Figure 11 :
Sketch (a) gives an example of two interacting features f and g. Feature g can either be moved to the left (b) or downwards (c). The decision variable d_{fg} determines which kind of movement has to be used for two-dimensional compaction.