Embedding Grids into Grids: Techniques for Large Compression Ratios

John A. Ellis

Department of Computer Science
University of Victoria
Victoria, British Columbia V8W 3P6, Canada

Mailing address:

John A. Ellis
Department of Computer Science
University of Victoria
P.O. Box 3055
Victoria
British Columbia V8W 3P6
CANADA

Electronic mail:

jellis@csr.uvic.ca

Abstract

We return to the problem of embedding 2-dimensional $(h \times w)$ guest grid graphs into 2-dimensional $(h' \times w')$ host grid graphs, where $w' < w$, and h' is the smallest integer such that $hw \le h'w'$. This 2-dimensional problem has many applications in computer science including the simulation of one grid of processors by another of different shape. Also, it can likely be applied to more complex problems such as those involving grids of higher dimension and hypercubes. It is already known that optimal dilation, namely two, is always obtainable whenever w/w', the *compression ratio*, is no greater than two. Only a restricted set of problem instances with compression ratios greater than two could previously be solved optimally, with respect to dilation. We introduce two new techniques, called *folding with compression* and *folding with extension*. By applying folding with compression, we show that optimal solutions always exist if $w' \ge 7h$ and the remainder of h' integer divided by h is no greater than h' integer divided by h. This condition includes all instances of the problem for which $w/w' \ge h \ge 7$, i.e., all instances for which the compression ratio and h are sufficiently large. There remain instances, among those of intermediate compression ratio, for which we still do not have optimal solutions. We show that some, but not all, of these can be solved by applying folding with extension.

This work was supported in part by a grant from the Natural Sciences and Engineering Research Council of Canada and was carried out at the Centre National de la Recherche Scientifique, Sophia Antipolis, France.

1. Introduction

We return to the problem of embedding 2-dimensional grid graphs into 2-dimensional grid graphs of smaller aspect ratio, previously considered in [1], [4] and [7]. A *graph embedding* is a one to one mapping from guest graph nodes to host graph nodes. The *dilation* of a guest edge is the length of a shortest path in the host between the images of the guest nodes incident to the edge. The dilation of an entire embedding is the largest dilation over all edges in the guest. The *expansion* of an embedding is the ratio of the number of nodes in the host to the number of nodes in the guest.

Consider a 2-dimensional grid of height h and width w, i.e., the graph comprising the node set $\{\{x,y\} \mid 1 \le x \le h, 1 \le y \le w\}$ and the edge set $\{\{\{u,v\},\{x,y\}\} \mid |u-x| + |v-y| = 1\}$. Let $h < w$ and let $s = \lceil (hw)^{0.5} \rceil$. For all w', $s \le w' < w$, let h' be the smallest integer such that $w'h' \ge wh$. We call w/w' the *compression ratio*. The question we ask is: Is there a dilation 2 embedding of the $h \times w$ guest grid into the $h' \times w'$ host grid for all compression ratios? We will refer to this problem as the *grid to grid embedding problem*.

The question was first raised in [1]. Subsequent work, [4] and [7], gave optimal solutions, dilation 2, for all instances of the problem with compression ratios no more than 2 and came close to optimal solutions for larger compression ratios. For compression ratios greater than 2, [7] gave dilation 2 with close to optimal expansion, and [4] gave dilation 3 into host grids of dimensions $(s+1) \times (s+1)$, and in general into grids of dimension $(h'+1) \times (w'+1)$, i.e., tending to optimal expansion for large graphs.

Examples of applications of the problem are: the simulation of one grid-like parallel processing architecture by another, see for example [6], and reconfiguring arrays of processors, see for example [5]. Several other applications are listed in [1], [4] and [7]. Solutions to this basic problem can be applied to more complex embedding problems, such as those involving multi-dimensional grids and hypercubes. For example, given that optimal dilation is obtainable for compression ratios ≤ 2, we can embed 2-dimensional grids into the smallest hypercube with sufficient nodes by compressing the grid into a host grid of width the next smaller power of 2. That ensures that the host grid is a subgraph of the hypercube and consequently that the guest grid can be embedded into the hypercube with dilation 2. See for example [2], [3].

Here we describe techniques whose application solves optimally many more instances of the problem, for compression ratios greater than 2. In particular we show that all instances of the problem with $h \geq 7$ and compression ratio $\geq h$ have optimal solutions. However, problem instances remain, among those with intermediate compression ratios, for which optimal solutions are still not known.

The following theorems summarise the new results. Let h, w, h' and w' be the dimensions of the guest and host grids, G and H, respectively, as just defined. Let **div** denote integer division and let h' **mod** h denote the remainder of h' integer divided by h,

Theorem 1

If $w' \geq 7h$ and $(h'$ **mod** $h) \leq (h'$ **div** $h)$ then G can be embedded into H with dilation 2.

Corollary 1

If $w/w' \geq h \geq 7$, i.e., h and the compression ratio are sufficiently large, then G can be embedded into H with dilation 2.

Theorem 2

If $(h'$ **mod** $h-1) \leq (h'$ **div** $h-1)$ and w' **div** h is odd with remainder zero then G can be embedded into H with dilation 2.

2. Overview

Figures 1, 2 and 3 illustrate the application of the primitive folding, compression and extension techniques to typical, large compression ratio, examples of the grid to grid embedding problem. Figures 1 and 2 show the embedding of a 7×289 grid into a 45×45 grid, with dilation 2. Expansion is also optimal since $45 \times 45 = 7 \times 289 + 2$, i.e., there are only 2 unused host nodes. Optimal results are obtained even though the dimensions of the guest and host do not conform to the constraints defined in Theorem 1, since $w' < 7h$. Hence there are instances of the problem outside of those defined by the constraints of the theorem for which the techniques still give optimal results.

In Figure 1 a) the guest has been divided into "segments" of varying widths, which total to 289. The shape of the segments has not been changed. Their relative positions and orientations indicate where each will be embedded into the square host. Of course the strategy is to fold the long, skinny grid around in a snake like manner so as to fit it into the short, fat grid, an idea which was introduced in [1].

Figure 1 b) shows how the shape of most segments has been changed by folding and compressing, so that they fit together into the 45×45 square. All the end segments, B, D, F, J, and N are folded through 180°. B and D are folded without compression, the input and output height remains 7. J and N are folds with compression, the input and output heights are increased to 8. F forms a transition from height 7 to the compressed height 8. Segments A, C and E suffer no deformation, I and K suffer unit compression, G, M, and O are compressed by 2. H, L and P are compressed by 1, 1 and 2 respectively so that they form a so called "tower" of uniform width.

****

* INSERT FIGURE 1 HERE *

****

Figure 2 illustrates the same example by drawing all the edges that were horizontal in the guest and showing how they are folded and compressed. Note that the endpoints of these edges "match" (with dilation ≤ 2) at the vertical joins between segments. The 2 unused host nodes are indicated by the 2 dots in the central segment at the bottom of the diagram.

****

* INSERT FIGURE 2 HERE *

****

Figure 3 illustrates a less complex instance, exemplifying Theorem 2. Here the guest is 5×60 and the host is 12×25, i.e., the expansion is exactly 1.0. The technique used is different from that of the previous example. The guest segments are extended rather then compressed. Thus for the folds, the input and output heights are reduced from 5 to 4, and likewise the heights of the internal segments. The diagram shows horizontal guest edges at the two folds, whereas for all the other segments vertical guest edges are shown. In general an extension along the horizontal axis is equivalent to a compression along the vertical.

****

* INSERT FIGURE 3 HERE *

****

We describe the folding, compression and extension techniques and how they can be fitted together to solve all instances of the problem within the constraints defined in the theorems. Section 2 reviews and extends the primitive compression patterns originally described in [4]. Section 3 shows how these compression patterns can be concantenated to solve the so called tower building problem, i.e., the non-uniform compression problem exemplified by segments H, L and P in Figure 1 a). Section 4 describes the folding techniques, with compression and with extension. Sections 5 and 6 show how to combine the primitive operations so as to solve the general problem. Theorem 1 and Corollary 1 are based on the application of folding with compression and are proved in Section 5. Theorem 2 is based on folding with extension and is proved in Section 6.

2. Compression Patterns

In this section we review and develop so called "compression tiles". The patterns defined by these tiles will allow us to perform the compression necessary for the central segments of the snaked guest grid. We find that we need tiles which reduce width by 1 and by 2, which we call 1-*tiles* and 2-*tiles* respectively. Also, in order to solve the tower building problem which is tackled in Section 3, we need even further modified tiles which have incomplete top and bottom rows. These we call *skewed* tiles.

2.1. Review of Unit Compression Tiles

[4] described a variety of unit compression patterns which in this paper we will call 1-*tiles*. These tiles are diagrams which define dilation two, expansion 1.0, embeddings from guest grids into host grids, in which the guest width is reduced by unity. For example, four variants of an 8×7 tile are shown in Figure 4. Note that these diagrams show the arrangement, after embedding, of the edges of the guest graph that had been horizontal in the guest. To verify that dilation is two, one has to look at all the horizontal edges and at all the edges that had been vertical in the guest and whose arrangement in the host is implied, but not shown, by the diagrams.

We want all our tiles to define embeddings which conform to the following conditions:

(1) The nodes from the left (right) column of the guest are embedded, with column order preserved, in the left (right) column of the host.

(2) All nodes in all host rows, except perhaps the top or bottom rows, are occupied by guest nodes.

(3) If the top or bottom host rows are only partially occupied, then, in these rows, all the occupied host nodes are contiguous and all the unoccupied host nodes are contiguous.

We will call these the *standard* conditions, embeddings which conform to these conditions will be called standard embeddings and the corresponding tiles will be called standard tiles.

Lemma 1

For all $n \geq 1$ there exists a dilation 2, standard embedding of the $n \times (n+1)$ grid into the $(n+1) \times n$ grid.

Proof See [4]. □

We will need to refer to the *orientation* of the verticals in a tile and to the *slant* of a tile. By the *orientation* of a vertical we mean the direction, *up* or *down*, implied by a left to right traversal of a guest row, at the point where the vertical is traversed. So, for example, the vertical at the top left of the *Tile c* in Figure 4 has orientation *down*, whereas the same vertical in *Tile d* has orientation *up*. We will call the direction of the path traced out by the verticals in a tile, which is either top left to bottom right or top right to bottom left, the *slant* of the tile. Thus the slant of *Tile c* in Figure 4 is top left to bottom right and that of *Tile a* is top right to bottom left. The four tile variations are obtained by combining the two possible orientations with the two possible slants.

Lemma 2

If any of the four 1-tile variations is placed side by side with any other 1-tile of the same size, then the dilation of any connecting edge across the vertical seam between the tiles is no more than two.

Proof Follows immediately from (1) above, i.e., the first standard condition, and the fact that the compression increases height by unity. □

We can create rows of tiles, with dilation two, in this way without limit. We will call this process the *horizontal concatenation* of tiles.

We are going to need tiles in which parts of the top and bottom host rows are unoccupied by guest nodes. We will call this type of tile a *skewed tile*.

Lemma 3

Let G be a grid of height h and width $w+1 = h+\delta+1$, where $1 \le \delta < h$ so that $h+1 \le w \le 2h-1$. Let H be a grid of width w and smallest height sufficient to contain G subject to the constraint imposed on the value of k below. For all k, $1 \le k \le w-1$, there exists a standard, dilation 2 embedding of G into H such that exactly k nodes of the top row of H are occupied by guest nodes.

Proof

By hypothesis, the host is large enough, even if $k = 1$. We observe that 1-tiles can be concatenated vertically, with dilation 2 across the horizontal seams between tiles, if the slants of adjacent tiles are opposing and if the orientations of the verticals in the bottom row of the upper tile and the top row of the lower tile are the same. Consider a pair of vertically concatenated, $(w+1)\times w$ 1-tiles. By removing $1 \le m \le w-1$ guest rows from the top of the pattern, we obtain a pattern in which m contiguous nodes of the the top row of the host are unoccupied, and hence where $w-1 \ge k \ge 1$ are occupied. Since $w > h$, at least h guest rows remain in the original two tile pattern. Hence we can then count down the remaining pattern until h guest rows have been accommodated.

The bottom host row in the resulting pattern may also be partially occupied, but it is clear that the occupied and unoccupied nodes will each be contiguous, as required by the standard conditions. □

See for example Figure 5 a) and b). Here an embedding of a 6×9 grid into an 8×8 with 4 host nodes in the top row occupied is derived from two standard 9×8 1-tiles.

2.2. Compression by Two

We also need standard tiles which reduce width by two. These tiles will be called *2-tiles*. The constructions are illustrated in Figures 6 - 10.

Lemma 4

For all $n \ge 1$ there exist:

(1) dilation 2, standard embeddings of $2n \times (4n+2)$ grids into $(2n+1)\times 4n$ grids,

(2) dilation 2, standard embeddings of $(4n+1)\times(4n+3)$ grids into $(4n+3)\times(4n+1)$ grids,

(3) dilation 2, standard embeddings of $(2n+1)\times(4n+4)$ grids into $(2n+2)\times(4n+2)$ grids,

(4) dilation 2, standard embeddings of $(4n+3)\times(4n+5)$ grids into $(4n+5)\times(4n+3)$ grids.

Proof

In each of the four cases we show solutions for $n = 1$ and $n = 2$. The way in which the construction is extended from 1 to 2 is elementary and clearly can be applied from n to $n + 1$ for any n.

(1) Figure 6 a) shows two solutions, the top two tiles in a tower, for $n = 1$. Figure 6 b) how the solutions for $n = 1$ can be extended to $n = 2$, and, by iteration, to any n.

(2) Figure 7 a) shows two solutions, the top two tiles in a tower, for $n = 1$. Figure 7 b) how the solutions for $n = 1$ can be extended to $n = 2$, and, by iteration, to any n.

(3) Figure 9 a) shows two solutions, the top two in a tower, for $n = 1$. Figure 9 b) how the solutions for $n = 1$ can be extended to $n = 2$, and, by iteration, to any n.

(4) Figure 10 a) shows two solutions, the top two in a tower, for $n = 1$. Figure 10 b) how the solutions for $n = 1$ can be extended to $n = 2$, and, by iteration, to any n. □

****

* INSERT FIGURES 6, 7, 8, 9 AND 10 HERE *

****

We note that the four cases just considered are the cases for which *host width* **mod** 4 = 0, 1, 2 or 3, (where **mod** is as defined in the introduction) i.e., all possible widths. We go on to construct *skewed 2-tiles*, which are analogous to the skewed 1-tiles.

Lemma 5

2-tiles can be vertically concatenated, without limit, with dilation 2.

Proof

In each of the four cases, two distinct tile solutions were given. That these can be subject to a vertical concatenation, with dilation 2 at the horizontal seam, is illustrated by Figures 6 through 10. Note that, in each of the four cases illustrated by the five figures, if the order (top, bottom) of the two tiles is reversed the dilation remains two. Consequently, the 2-tiles of the sizes that are illustrated can be vertically concatenated without limit. Finally we note that, in each of the four cases, and for both tile types one notes that, under expansion, the top and bottom guest rows remain unchanged in both tiles, so vertical matching is assured, no matter the tile size. □

We now show how to construct skewed 2-tiles from a pair of vertically concatenated 2-tiles. It is convenient to consider the cases of odd and even width separately. Note that Lemma 6 holds only for even k, whereas Lemma 7 holds for all k.

Lemma 6

Let G be a grid of height h and width $w+2 = h+\delta+2$, where $1 \leq \delta < h$. Let H be a grid of width w and height the smallest sufficient to contain G. If w is even, then for all even k, $2 \leq k \leq w-2$, there exists a dilation two, standard embedding of G into H such that exactly k nodes in the top row of H are occupied by guest nodes.

Proof

Let $w = 4n$ or $4n+2$ and consider a pair of vertically concatenated 2-tiles, respectively either $(2n+1) \times 4n$ or $(2n+2) \times (4n+2)$. For example consider Figure 6, which defines solutions for the case w **mod** 4 = 0, and Figure 9, which defines solutions for the case w **mod** 4 = 2. It is clear that, by removing j guest rows from the top of such a pair of tiles, we obtain a pattern in which $2j$ nodes in the remaining top host row are unoccupied. Hence k nodes are occupied, for even k in the range $2 \leq k \leq w-2$.

We may have to make a modification to the original pattern in order that the third standard condition be satisfied, i.e, that all occupied host nodes be contiguous, and likewise for the unoccupied. If the top guest row turns out to be of the type illustrated by host row 2 (counting down from the top) in either Figure 6, a) or b), then it must be replaced by a pattern as shown in the top guest row of Figure 6 c). The leftmost vertical in the original row must simply be moved right till it closes the gap with its neighbour.

Having started the pattern so as to achieve the correct k, we then count down h guest rows, in a sequence of vertically concatenated tiles of sufficient height, to achieve a complete solution. We may have to make a further modification to the original pattern. If the bottom guest row turns out to contain a gap, as for example row 2 of the lower tile in Figure 6 b), then it must be replaced by a pattern as shown in the bottom row of Figure 6 c). The gap must be closed in order that the third standard condition be satisfied. □

Figure 6 c) shows a skewed 2-tile which embeds a 5×10 grid into a 7×8 grid with 6 host nodes occupied in the top row. It is derived from a pair of 5×8 2-tiles.

Lemma 7

Let G be a grid of height h and width $w+2 = h+\delta+2$, where $1 \leq \delta < h$. Let H be a grid of width w and height the smallest sufficient to contain G. If w is odd, then for all k, $1 \leq k \leq w-2$, that leave sufficient space in H, there exists a dilation two, standard embedding of G into H such that exactly k nodes in the top row of H are occupied by guest nodes.

Proof

Consider the uppermost tiles in the pairs in Figures 7 and 10 again. If k is even, number the guest rows in these tiles 1, 2, ... , from bottom to top. Suppose we remove all guest rows from the top down to but not including guest row j, i.e., j rows remain. It is clear that $2j$ nodes in the topmost host row are occupied. As above, we may have to make a modification to the original pattern, in order to ensure that the embedding is standard. If the top guest row turns out to be of the type illustrated by guest row 3 (counting down from the top) in Figure 7 b), then it must be replaced by a pattern as shown in the top row of Figure 6 c).

If k is odd, number the guest rows 1, 2, ... , from top to bottom. Suppose we remove guest rows 1 through j. It is clear that guest row $j+1$ occupies $w-2j$ nodes in the uppermost of the used host rows. Again, the same modification defined in the previous paragraph may be necessary to ensure that the result is standard.

Having started the pattern so as to achieve the correct k, we then count down h guest rows, in a sequence of vertically concatenated tiles of sufficient height, to achieve a complete solution. We may have to make a modification to the original pattern. If the bottom guest row turns out to be of the type illustrated by guest row 4 (counting down from the top) in the lower tile of Figure 7 b), then it must be replaced by a pattern as shown in Figure 8 a). If the bottom guest row turns out to be of the type illustrated by row 5 in the lower tile of Figure 7 b) then it must be replaced by a pattern as shown in Figure 8 b). The same special cases arise in Figure 10, with the same solutions. □

3. Building Towers of 1-Tiles and 2-Tiles

The problem, which we have previously referred to as the *tower building* problem, must be solved as part of our general solution. Figure 11 illustrates an instance of the problem. Three independent segments are shown, two 7×11 segments and one 7×12. They are embedded into a 24×10 host, i.e., a tower of three 8×10 segments. Note that the dilation of this embedding is 2.

****

* INSERT FIGURE 11 HERE *

****

This example is in fact taken from segments H, L and P in Figure 1. The difficulty is of course that the top segment must be compressed by 2, the others by 1 and, since there is little extra space (only 2 spare host nodes) some kind of overlapping mechanism must be arranged between the individual embedded segments.

In its full generality we state the problem as follows. Consider a disconnected guest graph G consisting of $p+q$ components, comprising p *short blocks*, each an $h \times (w+1)$ grid, and q *long blocks*, each an $h \times (w+2)$ grid, where $h+1 \le w \le 2h-1$. Let the host H be a grid of height $(p+q)(h+1)$ and width w, i.e., a tower of $p+q$ blocks, each an $(h+1) \times w$ grid, and suppose $|H| - h < |G| \le |H|$, (where $|G|$ denotes the number of nodes in G), i.e., H is big enough to accommodate G but at least $w-h+1$ nodes in the bottom row of the host must be used. We find it convenient to refer to the value $|H| - |G|$ as the total underflow, denoted by *t_undfl*. We require that $0 \le t\_undfl < h$.

Let $w = h+\delta$. Then, from the limits we have put on w, $1 \le \delta \le h-1$. Since a short block contains ($hw+h$) nodes, a long block ($hw+2h$) nodes and a host block ($hw+w$), the underflow per short block is (w-h) = δ and the overflow per long block is $2h$-w = $h-\delta$. After say k guest blocks have been embedded, we can compare their size with the size of the corresponding k host blocks. If the size of the guests is less than the size of the hosts, then we will say that *underflow* exists. Otherwise we will say that *overflow* exists.

The solution has two parts, first we define the order in which the blocks should be embedded, then we use the techniques from Section 2 to do the embedding.

3.1. Ordering the Blocks

The two procedures *list_blocks_A* and *list_blocks_B*, place the guest blocks in a list which establishes an ordering. The first procedure ensures that underflow is always positive, but never as great as an entire host row. The second procedure ensures that overflow is always positive, but never as great as an entire host row. For both procedures, we prove certain properties which are necessary to show that the block order defined by the procedure is one which leads to a dilation 2 embedding and which will match adjacent tiles in the total embedding.

procedure list_blocks_A
{The procedure is to be used when: (both h and δ are even)
or (h is even and δ is odd) or (h is odd and δ is even).}

  $undfl$:= 0;
  while short or long overflow blocks remain **do**
    Add as many short blocks to the list as possible, subject to the conditions that:
    1. for each block added: $undfl := undfl + \delta$ and
    2. at the end of the additions, $undlf < h+\delta$.

    Add as many long blocks to the list as possible, subject to the conditions that:
    1. for each block added: $undfl := undfl - h + \delta$ and
    2. at the end of the additions, $undlf \geq 0$.
  endwhile
endprocedure

Lemma 8

The procedure *list_blocks_A* terminates, all blocks are placed in the list, at all points during the computation, $0 \leq undfl < h+\delta$ and if both h and δ are even then $undfl$ is always even.

Proof

To see that the procedure terminates and, consequently, that all blocks are placed on the list, we note first that both inner loops must terminate. Either the set of short (long) blocks must become empty, or the value of *undfl* must become greater (less) than the constant specified.

Suppose the outer loop does not terminate. If so, then some number of either short or long blocks (or both) remains invariantly.

Suppose only short blocks remain. Then, from the condition on the first inner loop, $undfl \geq h$, because each loop repetition adds δ so that if $undfl < h$, another short block would be added. But then, since only short blocks remain, it must be that $t\_undfl \geq h$, contradicting our problem definition.

Suppose only long blocks remain. Then, from the condition on the second inner loop, $undfl < h - \delta$. But then, since only long blocks remain, it must be that $t\_undfl < 0$, contradicting our problem definition.

Suppose both long and short blocks remain. Then, from the conditions on both the inner loops, $undfl \geq h$ and $undfl < h - \delta$, a contradiction, since $\delta \geq 1$.

To see that the value of *undfl* is always within the range specified, we note that it is initially zero and that no addition or subtraction is done unless the result remains within that range.

To see that the value of $undfl$ is always even, if both h and δ are even, we note that $undfl$ is initially zero and that it is changed only by the addition or subtraction of even numbers. □

procedure list_blocks_B;
{The procedure is to be used when h is odd and δ is odd.}
  $ovfl$:= 0;

  repeat
    Add as many long blocks to the list as possible, subject to the conditions that:
    1. for each block added: $ovfl := ovfl + h - \delta$
    2. at the end of the additions, $ovfl < h + \delta$

    Add as many short blocks to the list as possible, subject to the conditions that:
    1. for each block added: $ovfl := ovfl - \delta$
    2. at the end of the additions, $ovfl \geq 0$ and $ovfl$ is even;
  until no blocks were added at last pass through the loop;

  {At this point all long blocks must have been placed. See proof below}

  while short blocks remain **do**
    if *ovfl* < δ **then** $ovfl := ovfl + t\_undfl$ **endif**;
    $ovfl := ovfl - \delta$; Add a block to the list;
  endwhile
endprocedure

Lemma 9

The procedure *list_blocks_B* terminates, at entry and exit to the first inner loop *ovfl* is even, all blocks are placed in the list and, at all points during the computation, $0 \le ovfl < h+\delta$.

Proof

To see that the procedure terminates, we note first that both inner loops must terminate. Either the set of blocks must become empty, or the condition permitting further additions to the list must become false. The outer loop terminates since otherwise there must be an unlimited number of blocks.

To see that $ovfl$ is even at entry and exit to the first inner loop, we note that it is initially zero, that $h-\delta$ is even and so the first inner loop can not make $ovfl$ odd, and that the second inner loop is not entered unless the additions of short blocks will leave $ovfl$ even.

To see that all blocks are placed we first note that, at exit from the main loop, all long blocks have been placed on the list. This we deduce by the following argument. Suppose long blocks remain at exit from the main loop. Then, from the condition on the first inner loop, $ovfl \ge 2\delta$. Suppose short blocks also remain, then from the condition on the second inner loop, since $ovfl$ is even, either $ovfl < 2\delta$, since two blocks must be added to maintain evenness, or only one short block remains and it can not be added since it would leave $ovfl$ odd. If the first case we have a contradiction and in the latter case, since $t\_ovfl \le 0$, we must have that $\delta \ge ovfl+h-\delta$ because the underflow to be contributed by the one remaining short block must compensate for the current overflow plus that contributed by the remaining long blocks. But then $ovfl \le 2\delta-h$, contradicting our assumption that the first inner loop could not add more long blocks. Hence (1) if long blocks remain, no short blocks remain.

If no short blocks remain and some long blocks remain, then, since $ovfl \ge 0$ at exit, $t\_undfl < 0$, contradicting our problem definition. Hence (2) if no short blocks remain, then no long blocks remain. From (1) and (2) we deduce that, at exit from the main loop, no long blocks remain. Consequently, the final statement, which adds all remaining short blocks, is sufficient to ensure that all blocks are placed.

To see that the value of *ovfl* is always within the bounds specified, throughout the execution of the main loop, we note that it is initially zero, and that no change is made to $ovfl$ in the main loop, unless that change leaves the value within the range specified. At exit from the main loop no long blocks remain. The last statement adds any remaining short blocks to the list. If $ovfl$ is less than δ it is incremented by the value of t_undfl. Since $t\_undfl < h$, the result leaves $ovfl < h+\delta$. Since only short blocks are added, $ovfl$ can not fall below zero as this would imply that $t\_undfl < 0$, contradicting our problem definition. □

3.2. Embedding the Blocks in the Tower

We now show that the guest blocks can be embedded in the tower with dilation 2 and with another property, which will permit 1-tiles to abut the tower with dilation 2 at the vertical seams.

Lemma 10

There is a standard, dilation 2 solution to the tower embedding problem.

Proof

Suppose we embed the guest blocks in the order defined by the appropriate *list_blocks* procedure. At each point where a short or long block is to be embedded, the value of $undfl$ or $ovfl$ must be used to determine which skewed tile to use. There are four cases to consider.

Case 1: h is even and δ is even.

We have used *list_blocks_A*. We note that $w = h+\delta$, the width of the host block, is even, that both $h-\delta$ and δ are even and, by Lemma 6, $undfl$ is always even at the point where a long block is to be embedded. Since $undfl$ is the number of unoccupied host nodes in the host row currently being filled, Lemma 6 tells us that a suitable long block embedding is possible. By Lemma 2, short blocks can be embedded starting from any value of *undfl*.

Case 2: h is even and δ is odd.

We have used *list_blocks_A*. We note that $w = h + \delta$, the width of the host block is odd. Then Lemma 7 tells us that a suitable long block embedding is possible, no matter whether $undfl$ is odd or even at the point where a long block is to be added. Likewise, by Lemma 2, short blocks can be embedded starting from any value of *undfl*.

Case 3: h is odd and δ is even.

We have used *list_blocks_A* and $w = h + \delta$, the width of the host block, is odd. Hence, Lemma 7 tells us that a suitable long block embedding is possible, no matter whether $undfl$ is odd or even at the point where a long block is to be added. Likewise, by Lemma 2, short blocks can be embedded starting from any value of *undfl*.

Case 4: h is odd and δ is odd.

We have used *list_blocks_B*. We note that $w = h + \delta$, the width of the host block, is even. Since w is even, Lemma 6 tells us that a suitable embedding is possible, if $ovfl$ is even, at the point where a long block is to be added. But Lemma 9 assures us that indeed $ovfl$ is even at these points. By Lemma 2, short blocks can be embedded starting from any value of *ovfl*. Note that the last statement in *list_blocks_B* can adjust the value of $ovfl$ by adding t_undfl. Since the skewed tile chosen is dependent on the current value of $ovfl$, this adjustment implies that some unoccupied host nodes appear at this point in the pattern. □

The example in Figure 7 was constructed by applying *list_blocks_B*.

3.3. A Tower Embedding Property

We need to establish one more property of the tower embeddings. We are especially interested in the distribution of the guest row end points across the host row end points, because we must eventually show that the guest row end points will match their corresponding end points in adjacent 1-tiles.

We have designed the embedding procedure so that all of the nodes in guest block k, except perhaps some either from the top or bottom row, are embedded into nodes in host block k, i.e., the two sets of blocks do not get too far out of step. Suppose all nodes from guest block k are embedded into nodes of host block k, except that perhaps some nodes from the top guest row are embedded into the bottom row of host block $k-1$. We will call this a *top heavy* block embedding. Suppose all nodes from guest block k are embedded into nodes of host block k, except that perhaps some nodes from the bottom guest row are embedded into the top row of host block $k+1$. We will call this a *bottom heavy* block embedding. We will call an embedding which embeds all nodes from guest block k into host block k, a *neutral* embedding.

Consider again Figure 11 as an example. Three guest blocks are embedded into three host blocks. The top block is neutral and the remaining guest blocks are top heavy.

Lemma 11

The tower embedding process produces either top or bottom heavy, or neutral embeddings for all guest blocks.

Proof

Consider first the embeddings produced by using *list_blocks_A*. Lemma 8 assures us that $0 \le undfl < w$ before and after any block is embedded. Suppose $undfl > 0$, then for any guest block k, some of the first row of the guest block will be embedded into the last row of host block $k-1$ but no node of guest block k is embedded into host block $k+1$. If $undfl$ is zero, the next block embedding is neutral. Hence the ordering defined by *list_blocks_A* produces top heavy or neutral embeddings.

If *list_blocks_B* was used, then $0 \le ovfl < w$ at all times. Consequently, the ordering defined by *list_blocks_B* produces bottom heavy or neutral embeddings. □

4. Folding Patterns with Compression or Extension

The *folding* pattern described in [1] embeds any $n \times 2n$ grid into a $2n \times n$ grid so that the nodes in the leftmost column of the guest are embedded into the lower n nodes of the rightmost column of the host, with order preserved, and the nodes in the rightmost column of the guest are embedded into the upper nodes of the rightmost column of the host, with order preserved. The technique is illustrated in Figure 12 where the result of applying a fold to a 7×14 grid is shown. Clearly dilation is 2 and the pattern can be constructed for any n.

****

* INSERT FIGURE 12 HERE *

****

In this section we describe some more complex folding patterns which achieve folding and either compression or extension simultaneously. With these patterns, it becomes possible to work with host strips that are either thicker or thinner than the guest.

4.1. Folding with Compression

Figure 13 illustrates a dilation two embedding that achieves simultaneously a 90° bend and unit compression. A 4×5 grid is embedded into a 5×4 grid so that the leftmost column of the guest is embedded in the bottom row of the host and the rightmost column of the guest is embedded in the upper 4 nodes of the rightmost column of the host.

****

* INSERT FIGURE 13 HERE *

****

Lemma 12

For all $n \geq 1$ there exists a dilation two embedding of the $n \times n+1$ grid into the $n+1 \times n$ grid such that the nodes in the left column of the guest are embedded in the bottom row of the host and the nodes in the right column of the guest are embedded in the upper n nodes of the rightmost column of the host.

Proof

Figure 14 a) illustrates solutions for the base cases, $n=1$ and $n=2$. Figure 14 b) illustrates a technique which constructs a solution for $n+2$ from a solution for n. The box containing the open nodes represents an $(n+1)\times n$ solution. The addition of the black nodes around the entire periphery of the box represents a solution for $(n+3)\times(n+2)$. Consequently, a solution can be built for any n.

****

* INSERT FIGURE 14 HERE *

****

The construction goes as follows:

(1) Add nodes to the periphery of the solution of size n so as to construct an $(n+3)\times(n+2)$ grid.

(2) For all the new nodes in the bottom row, except the leftmost, add an edge connecting the node to its upper left neighbour. For example, in Figure 14 b), the edges drawn along the bottom.

(3) For all the new nodes in the rightmost column, except the topmost, add an edge connecting the node to its upper left neighbour. For example, in Figure 14 b), the edges drawn along the right side.

(4) For all remaining new nodes, add one or two edges connecting the node to its one or two next but one new neighbour(s) in the left column or top row. For example, in Figure 14 b), the edges drawn along the left side and along the top.

The edges just defined are the horizontal edges of the guest grid. It can be seen that none of these horizontal edges suffers dilation greater than two. We must take care with the vertical edges, whose position can be inferred. Is essential to specify which guest row is represented by which tile row.

(1) Assume that the two guest rows comprising the edges added at step 4 of the construction are adjacent in the guest. Note that the verticals between these rows suffer dilation no more than 2.

(2) Note that, in the first base case, if we assign the nodes in the rightmost guest column to the upper nodes of the rightmost host column *with order preserved*, then adding these two rows yields a dilation 2 solution for $n = 3$.

(3) Note that, in the second base case, if we assign the nodes in the rightmost guest column to the upper nodes of the rightmost host column *with order preserved*, then adding these two rows yields a dilation 2 solution for $n = 4$.

(4) Finally, consider the relationship between the current pair of rows and the previous pair, for example in Figure 14 b, rows *a, b, c* and *d*. Note that if a and c are adjacent rows in the guest, they match with respect to dilation of their verticals. Likewise for rows b and d. This is not the case if rows b and c are adjacent in the guest.

We can deduce from these observations that the following assignment of nodes from the rightmost guest column to the rightmost host column yields the required result. Let the nodes from the rightmost guest column be named 1, 2, 3 .... from bottom to top.

If n is odd, then their order in the host column, bottom to top is: 1, 2, 3, 5, 4, 6, 7, 9, 8 .... i.e., starting with (4,5), every other pair is reversed.

If n is even, then their order in the host column, bottom to top is: 1, 2, 3, 4, 6, 5, 7, 8, 10, 9, ... i.e., starting with (5,6), every other pair is reversed. Figure 14 b illustrates the row ordering for $n = 6$. □

We can construct two 180° folding with compression patterns from the 90° pattern just described. See Figure 15.

****

* INSERT FIGURE 15 HERE *

****

Lemma 13

For all $n \geq 1$ there exists a dilation two embedding of the $n \times (2n+1)$ grid into a $(2n+1) \times n$ grid such that the nodes in the rightmost column of the guest are embedded in the upper n nodes of the rightmost column of the host, and the nodes of the leftmost column of the guest are embedded into the lower n nodes of the rightmost column of the host. The order of the guest row end nodes is as specified in the proof of Lemma 12, except that in the bottom half of the tile this order is reversed.

Proof

Take a 90° tile of size n and create a second tile that is the mirror image of the first in a plane through the bottom row. Create a new tile by identifying each node in the bottom row of the first tile with its mirror image in the top row of the second tile. No edges appear in the new tile that were not present in the 90° components. Hence dilation remains two. □

Figure 15 a) illustrates the construction for $n = 4$. It will be used to connect the embedding of the guest into a host strip of height h to the embedding into a host strip of height $h+1$.

Lemma 14

For all $n \geq 1$ there exists an embedding of the $n \times (2n+2)$ grid into a $(2n+2) \times n$ grid such that the nodes in the rightmost column of the guest are embedded in the upper n nodes of the rightmost column of the host and the nodes of the leftmost column of the guest are embedded in the lower n nodes of the rightmost column of the host. The order of the guest row end nodes is as specified in the proof of Lemma 12, except that in the bottom half of the tile this order is reversed.

Proof

Take a 90° tile of size n and create a second tile that is the mirror image of the first in a plane through the bottom row. Create a new tile by adding an edge between each node in the bottom row of the first tile and its mirror image in the top row of the second tile. These horizontal guest edges appearing in the new tile are the only edges not present in the 90° components. Hence dilation remains two. □

Figure 15 b) illustrates the construction for $n = 4$. It will be used to connect the embedding of the guest into a host strip of height $h+1$ to the embedding into another host strip of height $h+1$.

4.2. Folding with Extension

Figure 16 illustrates a dilation two embedding that achieves simultaneously a 90° bend and unit extension. A 5×4 grid is embedded into a 4×5 grid such that the leftmost column of the guest is embedded with 4 nodes in the bottom row and one in the the next to bottom node of the left column of the host, and the rightmost column of the guest is embedded with 4 nodes in the rightmost column and one in the rightmost but one node of the top row of the host.

****

* INSERT FIGURE 16 HERE *

****

Lemma 15

For all $n \geq 2$ there exists a dilation two embedding of the $n+1 \times n$ grid into the $n \times n+1$ grid such that:

(1) the $n+1$ nodes in the left column of the guest are embedded, n in the leftmost nodes of bottom row of the host and one in the bottom but one node of the leftmost column of the host and

(2) the nodes in the right column of the guest are embedded, n in the rightmost column and one, from the second row, in the rightmost but one node of the top row of the host.

Proof

Figure 17 a) illustrates solutions for the cases, $n = 2$ and $n = 3$. Figure 17 b) illustrates a method that constructs a solution for $n+2$ from a solution for n. Consequently, a solution can be built for any $n \geq 2$. The box containing the open nodes represents an $n \times (n+1)$ solution. The addition of the black nodes aroung the entire periphery of the box represents an $(n+2) \times (n+3)$ solution.

The construction goes as follows:

(1) Add nodes to the periphery of the solution of size n to construct an $(n+2) \times (n+3)$ grid.

(2) For all the new nodes in the bottom row, except the two leftmost, add an edge connecting the node to its upper left neighbour. For example, in Figure 17 b), the edges along the bottom of the box.

(3) For all the new nodes in the rightmost column, except the two topmost, add an edge connecting the node to its upper left neighbour. For example, in Figure 17 b), the edges along the right of the box.

(4) Connect the leftmost but one node in the bottom row to its next but one vertical neighbour.

(5) Connect the topmost but one node in the rightmost column to its next but one left horizontal neighbour.

(6) For all remaining new nodes, add one or two edges connecting the node to its one or two next but one new neighbour(s) in the left column or top row. For example, in Figure 17 b), the edges along the left and the top of the box.

****

* INSERT FIGURE 17 HERE *

****

The edges just defined are the horizontal edges of the guest grid. It can be seen that none of these horizontal edges suffers dilation greater than two. We must take care with the vertical edges, whose position must be inferred. It is essential to specify which guest row is represented by which tile row.

(1) Note that for base case $n = 2$, the row labels 2 and 3 can be exchanged to yield a second dilation 2 solution. Likewise note that for base case $n = 3$, row labels 3 and 4 can be exchanged to yield a second dilation 2 solution.

(2) Assume that the two guest rows added at step 4 of the construction are adjacent in the guest. Note that the verticals between these rows suffer dilation no more than 2.

(3) Consider the pattern obtained by adding these two rows to the first base case, as defined in Figure 17 a). Assign the nodes in the rightmost guest column to the nodes of the rightmost host column, in the order 1, 3, 2, 4, 5, from the bottom up and with the last node (5) being mapped onto the rightmost but one node of the top row of the host. Alternatively, for the second row label assignment given for this base case, assign the guest nodes in the order 1, 2, 3, 5, 4. The result is a dilation 2

solution in both cases for $n = 4$.

(4) Consider the pattern obtained by adding these two rows to the second base case, as defined in Figure 17 a). Assign the nodes in the rightmost guest column to the nodes of the rightmost host column, in the order 1, 2, 4, 3, 5, 6, from the bottom up and with the last node (6) being mapped onto the rightmost but one node of the top row of the host. Alternatively, for the second row label assignment given for this base case, assign the guest nodes in the order 1, 2, 3, 4, 6, 5. The result is a dilation 2 solution in both cases for $n = 5$.

(5) Finally, consider the relationship between the previous 2 rows and the current pair being added, for example in Figure 17 b), rows 4, 5, 6 and 7. Note that if 6 and 5 were adjacent rows in the guest, they would match with respect to dilation of their verticals. Likewise for rows 4 and 7. This is not the case if rows 5 and 7 are adjacent in the guest.

We can deduce from these observations that the following assignment of nodes from the rightmost guest column to the rightmost host column yields the required result. Let the nodes from the rightmost guest column be named 1, 2, 3 .... from bottom to top.

If n is even, then their order in the host column, bottom to top can be: either 1, 3, 2, 4, 5, 7, 6, ... i.e., starting with (2,3), every other pair is reversed, where the last node in the list is mapped onto the rightmost but one node in the top row of the host, or 1, 2, 3, 5, 4, 6, 7, ... i.e., starting with (4,5), every other pair is reversed, where the last node in the list is mapped onto the rightmost but one node in the top row of the host.

If n is odd, then their order in the host column, bottom to top can be: either 1, 2, 4, 3, 5, 6, 8, 7, ... i.e., starting with (3,4), every other pair is reversed, where the last node in the list is mapped onto the rightmost but one node in the top row of the host. or 1, 2, 3, 4, 6, 5, 7, 8, ... i.e., starting with (5,6), every other pair is reversed, where the last node in the list is mapped onto the rightmost but one node in the top row of the host. Figure 17 b) illustrates a row ordering for $n = 6$. □

There are two 180° folding with extension patterns that can be constructed from the 90° pattern just described. See Figure 18 a) and Figure 18 b).

****

* INSERT FIGURE 18 HERE *

****

Lemma 16

For all $n \geq 2$ there exists a dilation two embedding of the $(n+1) \times 2n$ grid into a $2n \times (n+1)$ grid such that:

(1) the nodes in the rightmost column of the guest are embedded $n-1$ into the upper $n-1$ nodes of the rightmost column and one in the rightmost but one node of the top row of the host, and

(2) the nodes in the leftmost column of the guest are embedded $n-1$ into the lower $n-1$ nodes of the rightmost column and one in the rightmost but one node of the bottom row of the host.

The order of the guest row end nodes is as specified in the proof of Lemma 15, where the top and bottom halves of the tile must use the alternate row labellings specified.

Proof

Take a 90° tile of size n and create a second tile that is the mirror image of the first in a plane through the bottom row.

Choose one of the row labellings for the upper tile and the alternative for the lower tile.

Connect nodes with the same label across the horizontal seam.

The horizontal edges appearing in the new tile are the only edges not present in the 90° components and they suffer dilation no worse than two. Hence dilation remains two. □

Figure 18 a) illustrates the construction for $n = 4$. This pattern will be used to connect embeddings of the guest into adjacent host strips, each of height h-1.

Lemma 17

For all $n \geq 2$ there exists a dilation two embedding of the $(n+2) \times 2n$ grid into a grid of height $(2n+1)$ and comprising an $n \times (n+1)$ top half and an $(n+1) \times n$ bottom half such that:

(1) the nodes in the rightmost column of the guest are embedded $n-1$ in the rightmost column of the upper part of the host, and one, from the second row of the guest, in the rightmost but one node of the top row of the host, and

(2) the nodes in the leftmost column of the guest are embedded $n-1$ in the lower $n-1$ nodes of the rightmost column of the lower part of the host, and one, from the first row of the guest, in the rightmost but one node of the bottom row of the host.

The order of the guest row end nodes is as specified in the proof of Lemma 15, where either of the alternatives there defined may be chosen.

Proof

Take a 90° tile of size n and create a second tile that is the 90° anticlockwise rotation of the first.

Choose either of the two row labellings specified in the proof of Lemma 15 and apply it to both the upper and lower tile.

Connect nodes with identical row labels across the horizontal seam.

The horizontal edges appearing in the new tile are the only ones not present in the 90° components and they suffer dilation no worse than two. Hence dilation remains two. □

Figure 18 b) illustrates the construction for $n = 4$. This pattern will be used to connect the embeddings of the guest into host strips of height h and of h-1.

5. Embedding into Thick Strips

All the primitive techniques together with their necessary properties have now been described. It remains to show how they can be applied to give a general solution to our problem. In this section we describe a procedure *Embedding into Thick Strips* so as to justify Theorem 1. Figures 1 and 2, discussed in Section 2, illustrate the application of the procedure. An embedding of a 7×289 grid into a 45×45 grid is shown in some detail. Although the width of the host is only 6 tiles and a solution is not guaranteed to exist by Theorem 1, the techniques are still sufficient to create an optimal solution.

Procedure Embedding into Thick Strips

(1) Let *div* denote the integer division operator. Let r be the remainder of $h'\text{div}\,h$ and d be $h'\text{div}\,h$. Divide the host grid into horizontal strips, $(d - r)$ of height h, called *normal strips* and r of height $h+1$, called *thick strips*.

(2) Let $c = w - (d-r)w'$, that is, the number of guest columns that are not accommodated by the normal host strips. Let p be the maximum number of guest columns that can be accommodated by a thick host strip, without overflow, i.e., $p = \lfloor$ (number of nodes in a host strip) / (number of nodes in a guest column)$\rfloor = \lfloor w' \times (h+1)/h \rfloor$. Let $q = c - rp$.
Divide the guest grid into segments: $(d-r)$ of width w', the *normal* segments, $r-q$ of width p, the *short* segments, and q of width $p+1$, the *long* segments.
Divide each short guest segment into as many *regular blocks* of width $h+1$ as possible, leaving perhaps one block, the *overflow block*, of some width t, $h+2 \le t \le 2h$. Divide each long guest segment in a similar manner, except that the overflow block will be of width $t+1$.

(3) Embed the normal guest segments into the normal host strips, using the Standard fold at the ends of each strip, and of course with dilation one between the folds.

(4) For the remaining guest segments, let us consider the two possibilities, $q = 0$ and $q > 0$, separately.

Case 1: $q = 0$
Apply the fold with compression tile from Lemma 13 at the junction between the normal and thick strips, and the fold with compression tile from Lemma 14 at the other junctions. Compress each regular block, other than end blocks which are part of the fold tile, into its corresponding thick host strip by application of an $h+1 \times h$ tile. Fit the overflow block into an $(h+1) \times (t-1)$ host segment by using the upper h guest rows of a $t \times t-1$ tile.

Case 2: $p > 0$
Proceed as in Case 1, except for the handling of the overflow blocks. Now we have a total of r short and long guest segments to deal with, and hence, r overflow blocks of width t and of width $t+1$.
Embed the r overflow blocks into their corresponding r host blocks by applying the *tower building* of Section 3. Place the tower in the centre of the host strips.
For the blocks abutting the tower and for the blocks abutting the fold tiles, choose 1-tile variants that match their corresponding tower or fold tiles with dilation 2 at the vertical seams.

end_procedure

We wish to show that this procedure gives optimal results, on the specified instances of the problem. First let us assume that w is the largest integer such that $h'w' \ge hw$. We lose no generality since if w is not maximal in this sense, we can pad out the guest grid and solve the problem for the padded grid. Then we have the following lemma.

Lemma 18

Let t_undlf (the total underflow), denote the total size of the r host blocks (each $h+1 \times t-1$) minus the total size of all the guest overflow blocks. Then $h > t\_undfl \geq 0$.

Proof

The embedding procedure just described uses three types of components, the fold tiles, the 1-tiles and the compression of the overflow blocks. The first two embed guest segments into host segments of exactly the same size. We know by definition of the problem that the host grid is at least as large as the guest grid. Hence $t\_undlf \geq 0$.

Suppose $t\_underflow \geq h$. Then $h'w' - hw \geq h$, by the argument just given. But, by definition of the problem, w is the largest integer such that $h'w' \geq hw$. □

Lemma 19

$\lfloor p/(h+1) \rfloor = p - w'$ i.e., the number of blocks in a short or long guest segment, including a possible overflow block, is equal to the absolute amount of width reduction that must be applied to a short guest segment.

Proof

Suppose that $w' = kh + \delta$, for some k and δ, $0 \leq \delta < h$. Then $p = \lfloor (h+1)(k + \delta/h) \rfloor$ (from Step 2) and so $\lfloor p/(h+1) \rfloor = \lfloor \lfloor (h+1)(k+\delta/h) \rfloor /(h+1) \rfloor = k$.

But, from Step 2, $p - w' = \lfloor (w'h + w')/h \rfloor - w' = w' + \lfloor w'/h \rfloor - w' = \lfloor (kh+\delta)/h \rfloor = k$ □

Lemma 2 has already established that dilation across the vertical seams between a pair of 1-tiles is no more than 2, for all combinations of the four possible tile types. It remains to note that the dilation across vertical seams between tiles of different types is no more than 2.

Lemma 20

There exists a pair of 1-tiles that match the input and output of a fold with compression tile, with dilation 2.

Proof

All the n guest row ends of the top half of the fold tile are embedded in the upper n nodes of the end column of the host. The 1-tile that does likewise must give the required result, even if the order of some pairs of guest nodes is reversed. Similarly, all the n guest row ends of the bottom half of the fold tile are embedded in the lower n nodes of the end column of the host, and so a matching 1-tile exists. See Figure 19. □

Lemma 21

There exists a 1-tile that matches, with dilation 2, either top or bottom heavy tower tiles.

Proof

For a top heavy tile, choose the 1-tile that places all guest row ends in the upper host column nodes. For a bottom heavy tile, choose the 1-tile that places all guest row ends in the lower host column nodes. See Figure 20. □

We are now ready to argue that the procedure *Embedding into Thick Strips* does produce a dilation 2 embedding, so long as h and the compression ratio are large enough.

Theorem 1

Let $h \times w$ be the dimensions of the guest grid G and $h' \times w'$ be the dimensions of the host grid H. If $(h' \,\mathbf{mod}\, h) \leq (h' \,\mathbf{div}\, h)$ and $w' \geq 7h$ then there exists a dilation 2 embedding of G into H.

Proof

Let us consider each step in the procedure *Embedding into Thick Strips*

(1) We note that $(d-r)h + r(h+1) = dh + r = h'$, by definition of d and r.

(2) Clearly normal host strips and normal guest segments have the same number of nodes, short guest segments have no more nodes than a thick host strip and long guest segments, if there are any, have more nodes than a thick host strip.

(3) All the nodes in the normal host strips are used.

(4) *Case: q = 0*

Lemma 19 assures us that the width reduction obtained by reducing every one of the blocks defined by the procedure by unity, is exactly the amount necessary to reduce the length of a short guest segment to w'. Since there are no long guest segments, every short guest segment fits into its corresponding host strip without overflow. To assure ourselves that dilation is no more than two it is only necessary to note that the tiles used (1-tiles, and fold with compression) can match at the vertical seams with dilation no worse than two by Lemmas 2 and 20.

Case: q > 0

Lemma 18 and the arithmetic in step 2 assure us that the parameters for the tower building problem are valid and hence Lemma 10 applies. Lemma 10 assures us that the set of overflow blocks, of widths t and t plus one, can be embedded into the corresponding tower of r host blocks, with dilation two. Since $w' \geq 7h$, there are at least 7 tiles along the width of the host strip, including the 2 tiles part of the folds. Suppose there are exactly 7 tiles. We choose to order them: a fold, two 1-tiles, a tower tile, two 1-tiles, fold. Lemmas 2, 20 and 21 assure us that dilation at the vertical seams is no more than 2, if the correct choice of 1-tile is made.

If there are more than 7 tiles across the width of the host strip, Lemma 2 assures us that further 1-tiles can be inserted between the 1-tiles defined for the base case. □

Corollary 1

If $w/w' \geq h \geq 7$, i.e., h and the compression ratio are sufficiently large, then G can be embedded into H with dilation 2.

Proof

Since $hw \leq h'w'$, $h \leq w/w' \Rightarrow h \leq h'/h \Rightarrow h \leq \lfloor h'/h \rfloor$. But $r < h$ and so $r \leq \lfloor h'/h \rfloor$.

Since $h'/h \geq w/w'$, $w/w' \geq h \Rightarrow h' \geq h^2$. But by problem definition $w' \geq h'$. Hence $w' \geq h^2$. Hence

$h \geq 7 \Rightarrow w' \geq 7h$.

From Theorem 1, if $w' \geq 7h$ and $r \leq \lfloor h'/h \rfloor$ then G can be embedded into H with dilation 2. □

6. Embedding into Thin Strips

Folding with compression permits us to construct optimal solutions for the grid to grid embedding problem, so long as h and the compression ratio are large enough. Many, but not all, of the problem instances with intermediate compression ratios, can be solved either by techniques so far described, or by our second technique, *Embedding into Thin Strips*. This is accomplished by using *fold with extension* patterns, which were defined in Section 4. Now one embeds the guest segments into the thin host strips by extension rather than by compression. This can be accomplished by using 1-tiles rotated through 90°. The process is illustrated in Figure 3, considered in Section 2. A straightforward application of folding with extension, that does not attempt an analogue of "tower building", yields the following theorem.

Theorem 2

Let G and H be the guest and host grids of the grid to grid embedding problem. If $(h' \textbf{ mod } (h-1)) \leq (h' \textbf{div} (h-1))$ and $w' \textbf{ div } h$ is odd with no remainder then G can be embedded into H with dilation 2.

Proof

Since $h' \textbf{ mod } (h-1) \leq h' \textbf{ div } (h-1)$ the height of the host can be divided into strips, some of height h and some of height $h-1$, the thin strips. The fold transition from normal to thin strips can be accomplished by the construction in Lemma 17. The fold transition between thin strips can be accomplished by the construction in Lemma 16.

We observe that the fold tile embeds the end points of the top two guest rows in the top host row on input and the bottom host row on output and that the order on input is the reverse of the order on output. Consequently the top and bottom halves of the fold can be matched by two 1-tiles of opposing slopes and orientation. See Figure 3.

Since w' is evenly divisible by h, a sequence of 1-tiles rotated through 90° exactly fills the width of the host. Adjacent 1-tiles across the width of the host can match if they have opposing slants, since we are in fact using vertical concatenation of standard tiles. Since the number of 1-tiles across the width of the host is odd, the slants of left and rightmost 1-tiles are the same. Hence they can match the fold tiles that they abut, because if the fold tile on the left of the row is the upper half of a complete fold, then the fold tile on the right is a lower half, and *vice versa*. See Figure 3. □

Figure 3, described in detail in Section 2, illustrates the application of embedding into thin strips to the problem of embedding a 5×60 grid into a 12×25 grid.

7. Conclusions

Previous work has solved the grid to grid embedding problem with optimal dilation for all instances with compression ratio no greater than 2. In this paper we have shown that optimal solutions also exist for all sufficiently large compression ratios and $h \geq 7$. We conjecture that it will be easy to cover the cases with $h < 7$ and we note that the techniques described here and in previous work cover many instances with intermediate compression ratio. Nevertheless, there exist instances of intermediate compression ratio which do not seem to be solved by any technique so far seen. For example the case of guest grid 7×43 into host 17×18 appears to be intractable at the moment. The remaining challenge is thus to find ways of handling these outstanding instances, or to show that there exist instances with no optimal solution. One conjectures that the scope of the intractable instances has now been so reduced that it may be possible to establish that all instances have optimal solutions.

Acknowledgements

The author wishes to thank his colleague Frank Ruskey for helping work out the construction of the fold patterns and the referee for a careful reading and constructive suggestions. The author is greatly indebted to the Centre National de la Recherche Scientifique, Sophia Antipolis, France, who provided the facilities for this work.

References

[1] Aleliunas, R. and Rosenberg, A. L., ‘‘On Embedding Rectangular Grids in Square Grids’’, IEEE Trans. on Computers, C-31, 9, pp. 907 - 913, 1982.

[2] Bettayeb, S., Miller, Z. and Sudborough, I. H., ‘‘Embedding Grids into Hypercubes’’, Jnl. of Computer and System Sciences, 45, 3, pp. 340 - 366, 1992.

[3] Chan, M. Y., ‘‘Embedding of Grids into Optimal Hypercubes’’ SIAM Journal on Computing, 20, 5, pp. 834 - 864, 1991.

[4] Ellis, J., ‘‘Embedding Rectangular Grids into Square Grids’’, IEEE Trans. on Computers, 40, 1, pp. 46 - 52, 1991.

[5] Lombardi, F., Sciuto, D. and Stefanelli, R., ‘‘An Algorithm for Functional Reconfiguration of Fixed-Size Arraysρq, IEEE Trans. on Computer-Aided Design, 10, pp. 1114 - 1118, 1988. Proc. Int. Conf. on Parallel Processing, 1987.

[6] Ma, Y. E. and Shea, D. G., ‘‘The Embedding Kernel on the IBM Victor Multiprocessor for Program Mapping and Network Reconfiguration’’, Proc. Second IEEE Symposium on Parallel and Distributed Processing, pp. 874 - 879, 1990.

[7] Melhem, R. G. and Hwang, G., ‘‘Embedding Rectangular Grids into Square Grids with Dilation Two’’, IEEE Trans. on Computers, 39, 12, pp. 1446 - 1455, 1990.