3-dimensional Channel Routing

ATTILA REISS

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
reiss@cs.bme.hu

DÁVID SZESZLÉR

Department of Computer Science and Information Theory
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
szeszler@cs.bme.hu

Abstract: Consider two parallel planar grids of size $w \times n$. The vertices of these grids are called terminals and pairwise disjoint subsets of terminals are called nets. We aim at routing all nets in a cubic grid between the two layers holding the terminals. However, to ensure solvability, it is allowed to introduce an empty row/column between every two consecutive rows/columns containing the terminals (in both grids). Hence the routing is to be realized in a cubic grid of size $2n \times 2w \times h$. The objective is to minimize the height h. In this paper we generalize previous results of Recski and Szeszlér [10] and show that every problem instance is solvable in polynomial time with height $h = O(\max(n, w))$. This linear bound is best possible (apart from a constant factor).

Keywords: VLSI layout, 3-D routing

1 Introduction

Traditionally, the detailed routing phase of the design of VLSI (Very Large Scale Integrated) circuits was considered as a **2-dimensional problem**, gradually extended to $2, 3, \ldots$ layers. Even within this problem, single row routing and channel routing are the better understood subproblems, where the terminals are placed on one side, or two opposite sides, respectively, of a rectangular circuit board, and the routing is to be realized on a few planar layers. Since the length n of the board is fixed by the row(s) of terminals and the number of layers k is fixed, the objective is to minimize the *width* w of the routing.

The specification of a (2-dimensional) channel routing problem instance involves a family of pairwise disjoint subsets of the terminals, called *nets*. By a *routing* we mean an assignment of pairwise vertex-disjoint Steiner-trees in the 3-dimensional grid (of size $w \times n \times k$) to each net, such that the assigned tree connects the terminals of the corresponding net.

Out of the wide literature on detailed routing, we are going to use the following result on 2-layer channel routing.

Theorem 1 (A. Recski and F. Strzyzewski, [9]) *Every channel routing problem with length l can be solved in $O(l)$ time on 2 layers such that for the width w of the obtained routing $w \leq \frac{3}{2}l$ holds. Furthermore, if each net contains two terminals only, one on the upper, and one on the lower boundary then the bound on the width can be improved to $w \leq l$.*

In single row routing and channel routing the inputs are essentially one-dimensional (one or two lists of terminals) and the output is essentially two-dimensional (a fixed number of planar layers). However, as technology permits more and more layers, a "real" **3-dimensional** approach becomes reasonable. The research of 3-dimensional routing started in the 1980s and there are plenty of deep results in this area, see [1, 2, 3, 4, 5, 6, 7, 8, 12, 13], for example. However, most of these results embed certain "universal-purpose" graphs (like n-permuters, n-rearrangeable permutation networks, shuffle-exchange graphs) into the 3-dimensional grid, ensuring that *pairs* of terminals can be connected, moreover, in some papers along *edge-disjoint* paths.

In [10] the *single active layer routing problem* (abbreviated as *SALRP*) is considered, which can be regarded as a 3-dimensional analogue of single row routing. Here the terminals are placed on a single planar grid of size $w \times n$ and the third dimension (above this grid, with height h) is for interconnections only. *Multiterminal nets* are also allowed and the interconnection of the terminals within each net is to be realized along *vertex-disjoint* paths (or Steiner-trees). One can easily see even in small instances like 2×2 or 4×1 that a routing is usually impossible (with an arbitrary height). Therefore it is allowed to extend the length and the width of the grid to $w' = sw$ and $n' = sn$, where the *spacing* s is a fixed integer. This is done by introducing $s - 1$ pieces of empty rows and columns between every two consecutive rows and columns containing the terminals. We are going to use the following result:

Theorem 2 (A. Recski and D. Szeszlér, [10]) *If $s \geq 2$ then every SALRP instance can be solved with height $h = 6\max(n, w)$ in $O(t(w + n))$ time, where t is the number of nets.*

We are going to use the following property of the construction presented in the proof of the above theorem: for each terminal t the vertical (that is, parallel with the height) line of the grid intersecting t is occupied by a single "long" vertical wire segment (which, of course, ends in t). (We also mention that the above bound of $6\max(n, w)$ can be improved if either the value of s is increased or the number of terminals in a net is limited to 2; see [10] for the details.)

In this paper we concentrate on a generalization of the single active layer routing problem. In the *3-dimensional channel routing problem* (or *3DCRP* for short) the terminals are placed on two parallel grids of size $w \times n$. Hence this problem can be regarded as a 3-dimensional analogue of channel routing. The 3DCRP problem is of interest not only from a technical point of view (see [4] for example), but also in a theoretical sense: in constrast to the essential difference in complexity between (2-dimensional) single row routing and channel routing, there does not seem to be such a difference between their 3-dimensional analogues.

We adopt the notion of spacing for 3-dimensional channel routing in a straightforward way: $s - 1$ pieces of empty rows/columns are inserted between each two original rows/columns containing the terminals (on both grids). Thus the routing is to be realized in a cubic grid of size $(s \cdot n) \times (s \cdot w) \times h$, where the *height* h is to be minimized. We show that if $s \geq 2$ then every *3DCRP* instance can be solved in polynomial time with height $h = 15\max(n, w)$ and this bound can be improved to $h = 3\max(n, w)$ if each net contains two terminals only, one on the bottom and one on the top grid.

2 Definitions and main results

Assume that two parallel grids of size $w \times n$ are given. The vertices of these grids are called *terminals.* A *net* N is a set of terminals. A *3-dimensional channel routing problem* (or *3DCRP* for short) is a set $\mathcal{N} = \{N_1, N_2, \ldots, N_t\}$ of pairwise disjoint nets. n and w are the *length* and the *width* of the routing problem, respectively. By a *bipartite 3DCRP* we mean the special case of the 3DCRP where each net consists of two terminals only, one on the bottom and one on the top grid.

By a *spacing of s_w in direction w* we are going to mean that we introduce $s_w - 1$ pieces of extra columns between every two consecutive columns (and also to the right hand side of the rightmost column) of the original grid. This way the width of the grid is extended to $w' = s_w \cdot w$. A *spacing of s_n in direction n* is defined analogously.

A *solution with a given spacing s_w and s_n* of a routing problem $\mathcal{N} = \{N_1, N_2, \ldots, N_t\}$ is a set $\mathcal{T} = \{T_1, T_2, \ldots, T_t\}$ of pairwise vertex-disjoint Steiner-trees in the cubic grid of size $(w \cdot s_w) \times (n \cdot s_n) \times h$ (between the two parallel planar grids containing the terminals) such that the terminal set of T_i is N_i for every $1 \leq i \leq t$. The Steiner-trees T_i are called *wires*. The *height* h of the routing is to be minimized.

In order to simplify the description of the routings, let us work in the (x, y, z) coordinate system (see Figure 1). The routing should be realized in a box of size $(s_w \cdot w) \times (s_n \cdot n) \times h$. We are going to use the term $z = h_0$ *h-plane* to refer to all the vertices with $z = h_0$ (this layer is a cross-section of the cubic grid perpendicular to the height, therefore of size $(s_w \cdot w) \times (s_n \cdot n)$). The meaning of the terms *n-plane* and *w-plane* is analogous.

Assume that the $z = 0$ h-plane is the bottom grid, and the $z = h$ h-plane is the top grid. The terminals are situated on these two grids, on each grid at most $n \cdot w$ terminals. The x and y coordinates of the terminals are divisible by s_n and by s_w, respectively.

Let us introduce the term *w-line*, a line that is parallel with the width w of the grid. The meaning of the terms *n-line* and *h-line* is analogous. Moreover, we are going to use the term *w-wire segment* to refer to a wire segment that belongs to a w-line of the grid. The meaning of the terms *n-wire segment* and *h-wire segment* is analogous.

Terminals of the bottom grid with a common x-coordinate are going to be referred as a *row of terminals*. For example, the terminals $b_1, b_2, \ldots, b_w$ of Figure 1 form the *$x = a$ row of terminals*. The meaning of the term *$y = b$ row of terminals* on the top grid is analogous. This includes n pieces of terminals (see $q_1, q_2, \ldots, q_n$ in Figure 1).

Lemma 3 *If $s_w \geq 2$ and $s_n \geq 4$ then every 3DCRP can be solved with height $h = 6\max(n, w)$ in polynomial time.*

PROOF: Let us move the terminals on the top grid by two units in direction of y and then project these terminals on the bottom grid. This way we obtain a SALRP with $s_w, s_n \geq 2$, which can be solved with height $h = 6\max(n, w)$ by Theorem 2. Since each h-line above a terminal is occupied by a single h-wire segment (see the remark after Theorem 2), this solution can be modified in a straightforward way to give a solution of the original 3DCRP. □

Because of the above lemma, in this paper we restrict ourselves to the case of $s_w = s_n = 2$; that is, only one extra row and column is inserted between every two consecutive rows/columns containing the terminals. The main results of this paper are the following two theorems.

Theorem 4 *Every 3-dimensional channel routing problem can be solved with $s_w = s_n = 2$ and height $h = 15\max(n, w)$.*

Theorem 5 *Every bipartite 3-dimensional channel routing problem can be solved with $s_w = s_n = 2$ and height $h = 3\max(n, w)$.*

The following lemma shows that the bounds of Theorems 4 and 5 are best possible apart from a constant factor.

Lemma 6 (A. Recski and D. Szeszlér, [10]) *For any given n there exists a single active layer routing problem that cannot be solved with height h smaller than $\frac{n}{2s_w}$.*

For a (simple) proof the reader is referred to [10]. (We mention that a straightforward modification of the proof presented in [10] shows that the lower bound of the above lemma can be improved to $\frac{n}{s_w}$ in case of the 3-dimensional channel routing problem.)

PROOF OF THEOREM 4 FROM THEOREM 5: Assume that a (general) 3DCRP instance is given (with $s_w = s_n = 2$). To obtain a routing, first solve the bottom and the top grid as two separate SALRP instances. This can be done with height $h_1 = 6\max(n, w)$ for both the top and the bottom grid by Theorem 2. Now consider, for example, the bottom grid and choose a terminal from each net (that also has a terminal on the top grid) arbitrarily. Since in the solution of the SALRP each h-line intersecting a terminal is used by a single h-wire segment (see the remark after Theorem 2), each chosen terminal can be connected with the corresponding vertex of the $h = h_1$ h-plane without ruining the SALRP solution. Repeating the same process for the top grid, we obtain a bipartite 3DCRP (to be routed between the two SALRP solutions), which can be solved with height $h = 3\max(n, w)$ by Theorem 5.

Hence the given 3DCRP instance is solved and the required height is $h = 2 \cdot 6\max(n, w) + 3\max(n, w) = 15\max(n, w)$. □

So further on in this paper we restrict ourselves to the bipartite 3DCRP.

3 Proof of Theorem 5

3.1 An overwiew of the main steps

Assume that a bipartite 3DCRP instance is given. Denote a net by $N_k = \{b_k, t_k\}$, where b_k is a terminal on the bottom grid, and t_k is a terminal on the top grid. The number of rows (w) and columns (n) is given. Consider the $x = a$ row of terminals on the bottom grid, the terminals are $b_1, b_2, \ldots, b_w$ (see Figure 1). The corresponding terminals on the top grid are $t_1, t_2, \ldots, t_w$.

Let us introduce the terms *access line*, *first virtual terminal* and *second virtual terminal.* Consider the $y = b$ row of terminals on the top grid. This means n pieces of terminals ($q_1, q_2, \ldots, q_n$, see Figure 1). Assign an n-line, called the *access line*, on the $y = b + 1$ w-plane to each net (see the next section for the details). For example, in Figure 1 we assigned the access line e_i to the net $N_i = \{b_i, t_i\}$. This access line is situated in the $y = b + 1$ w-plane, and let the z-coordinate be $z = h_e$. The *first virtual terminal* is on this access line, with $x = a$, $y = b + 1$ and $z = h_e$ coordinates. Let us denote this first virtual terminal by f_i (so it belongs to the net $\{b_i, t_i\}$).

We also assign a *second virtual terminal* to each net, situated in the $z = h_o$ h-plane, above the terminals on the bottom grid. For example, we assign the second virtual terminal s_i to the net $N_i = \{b_i, t_i\}$ (see Figure 1). The x-coordinate of s_i equals to the x-coordinate of b_i, and the y-coordinate of s_i is arbitrary (see Section 3.3).

The solution of the bipartite 3DCRP will be broken up into three main phases:

Phase I. The wiring between the terminals on the top grid (t_i) and the corresponding first virtual terminals (f_i).

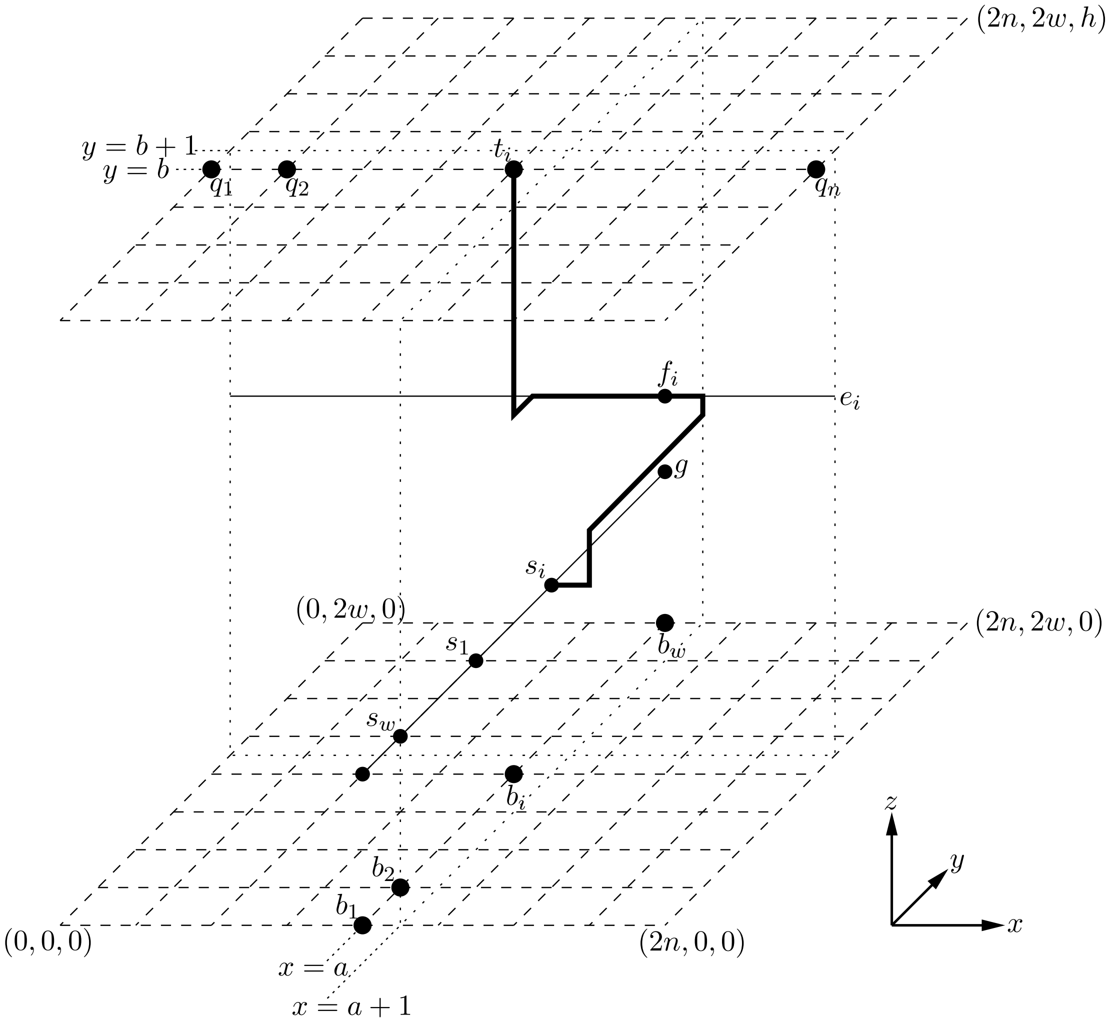

Figure 1

Phase II. The wiring between the first virtual terminals (f_i) and the second virtual terminals (s_i).

Phase III. The wiring between the second virtual terminals (s_i) and the terminals on the bottom grid (b_i).

We mention, that in Figure 1 only the first two phases are shown (so only the wiring between the terminals on the top grid and the second virtual terminals).

3.2 Phase I

First we have to solve the wiring between the terminals on the top grid (t_i) and the first virtual terminals (f_i). This should be as following (see Figure 1): we introduce an h-wire segment from the terminal t_i to reach the height h_e of the access line e_i. Further, we connect this h-wire segment with the access line e_i by a 1-unit w-wire segment. Finally, we connect this w-wire segment with the first virtual terminal f_i by an n-wire segment on the access line e_i. It is easy to verify that if we wire all the terminals on the top grid with the corresponding first virtual terminal, than the h-, n- and w-wire segments of the routings cannot cross the routing of any other net (provided that condition (1) below is true).

Note that the x-coordinate of the first virtual terminals already equals the x-coordinate of the corresponding terminal on the bottom grid, for example the x-coordinate of f_i is $x = a$.

Now, let us see, how to assign the access lines to the nets. To do this, we will have to consider two conditions:

- (1) If the upper terminals (on the top grid) of two nets have the same y-coordinates, then the access lines belonging to these two nets must be at different heights (so the access lines must be situated in different h-planes). For example, the n pieces of nets containing the terminals on the top grid with $y = b$ coordinate $(q_1, q_2, \ldots, q_n)$ have an access line on the $y = b + 1$ w-plane, and these n pieces of access lines are different, so they are necessarily at different heights.

- (2) If the lower terminals (on the bottom grid) of two nets have the same x-coordinates then the access lines belonging to these two nets must be at different heights. For example the access lines belonging to the nets $\{b_1, t_1\}$,$\{b_2, t_2\}$,. . .,$\{b_w, t_w\}$ must be at different heights.

In order to fulfil the conditions (1) and (2), we define a graph G in the following way: the vertices of G are the columns on the bottom grid (only the columns with terminals, so this means n vertices in the graph) and the rows on the top grid (this means w vertices in the graph), and the edges should correspond to the nets. Since each net contains two terminals, one on the bottom and one on the top grid, therefore in G each edge has a vertex in the set of lower columns and one in the set of upper rows. So G is a bipartite graph. The degree of the vertices representing a lower column is at most w, and the degree of the vertices representing the upper rows is at most n.

After all, we have now an edge-colouring problem, since an assignment of appropriate heights to the access lines (considering the two conditions above) corresponds to a good edge-colouring of the graph G. Since G is a bipartite graph, the chromatic index χ' of G is at most $\max(n, w)$. Assume that $n \geq w$, so $\chi' \leq n$.

Now, to all nets we have determined the position of the access lines, since each colour class defines a height. It was mentioned before that for example the y-coordinate of the access line belonging to the net $\{b_i, t_i\}$ is $y = b + 1$, and now the colour class of this net defines the $z = h_e$ coordinate of the access line too.

The problem of the wiring between the terminals on the top grid and the first virtual terminals is thus solved. The next problem is to solve the wiring between the first virtual terminals and the second virtual terminals.

3.3 Phase II

Under the top grid, there are n pieces of h-planes containing the access lines. Let us introduce an extra h-plane between every two consecutive h-planes (and also one under the lowest h-plane). These extra h-planes will be necessary for the wiring between the first virtual terminals and the second virtual terminals.

Further, let us consider only the $x = a$ and $x = a + 1$ n-planes from the grid of size $(2 \cdot w) \times (2 \cdot n) \times h$ (see Figure 2). We chose $w = 6$ for simplicity in Figure 2. As shown in Figure 2, the terminals on the bottom grid $(b_1, b_2, \ldots, b_w)$ and the corresponding first virtual terminals $(f_1, f_2, \ldots, f_w)$ and second virtual terminals $(s_1, s_2, \ldots, s_w)$ have all $x = a$ coordinates, while the x-coordinate of the terminals on the top grid $(t_1, t_2, \ldots, t_w)$ is arbitrary. For example, in Figure 2 the terminal t_2 has an x-coordinate of $x = a$, and $x \neq a$ holds for the x-coordinate of the other terminals t_i on the top grid.

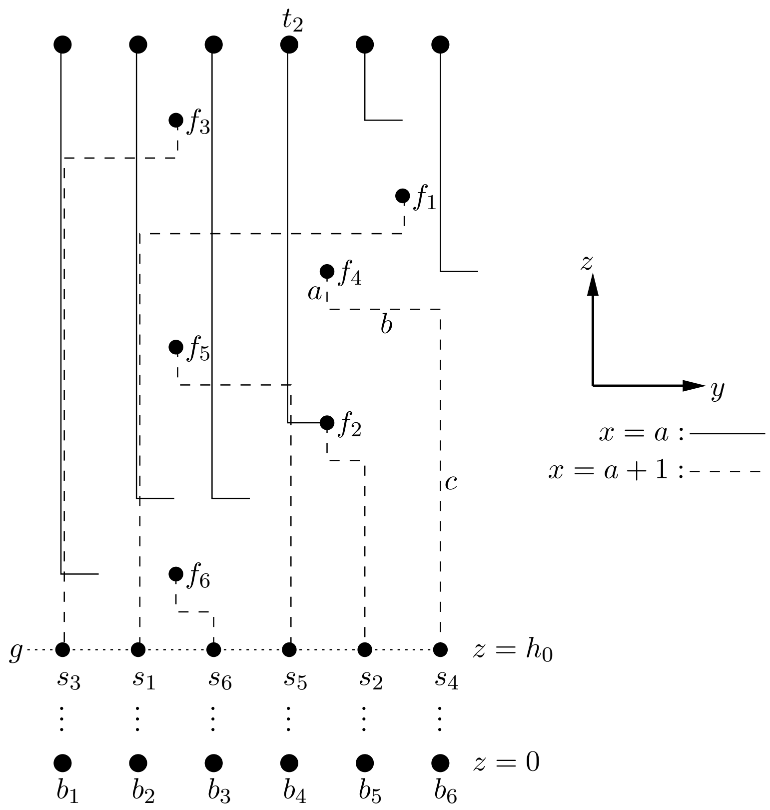

Figure 2

For the wiring between the first and second virtual terminals, we will use two consecutive n-planes, the $x = a$ and $x = a+1$ n-planes. In Figure 2 wire segments of the $x = a$ and $x = a+1$ n-planes are represented with continuous and dashed lines, respectively. Figure 2 shows w (here $w = 6$) pieces of continuous L-shaped wire segments. These wire segments belong to the routings between the terminals on the top grid and the first virtual terminals. The number of these wire segments is w, because on the top grid, w pieces of terminals belong to the $x = a$ row of terminals. In Figure 2, the x-coordinate of the terminal t_2 was originally $x = a$, so the access line belonging to this terminal was not used, because the first virtual terminal also has $x = a$ coordinate. The other L-shaped wire segments do not belong to the terminals $t_1, t_2, \ldots, t_6$, so the corresponding first virtual terminals do not have $x = a$ coordinate.

The z-coordinates of the first virtual terminals $f_1, f_2, \ldots, f_w$ are pairwise different according to condition (2) above. Otherwise there is no other restriction for their position: the w pieces of first virtual terminals can use any w heigths of the possible n (recall that $n \geq w$ was assumed).

The second virtual terminals are situated on the line g in Figure 2, above the terminals $b_1, b_2, \ldots, b_w$. This line is situated in the $z = h_0$ h-plane, as shown in Figure 1. On the line g, there are w pieces of free places for second virtual terminals, but it will be clear only later which second virtual terminals gets in which place.

On the $x = a + 1$ n-plane there are not any wire segments yet, only the intersection points of this n-plane and the access lines are occupied. So, for the wiring between the first virtual terminals and the second virtual terminals we will use this n-plane.

Arrange the first virtual terminals f_i according to the ascending order of their z-coordinates (on Figure 2 this order is $f_6, f_2, f_5, f_4, f_1, f_3$). We proceed in this order. The wiring should be realized in the following way: consider first the lowest one of the first virtual terminals f_i (f_6 in Figure 2). Going either to the left or to the right, place the corresponding second virtual terminal s_i to the first free place on the line g (in Figure 2, we chose going right). Further, the routing should be realized with three wire segments, called *wire segments of type a,b and c* (see the routing between f_4 and s_4 on Figure 2). a is a 1-unit h-wire segment, b is a w-wire segment connecting the columns of f_i and s_i and c is again an h-wire segment that ends in s_i. After this, consider the next first virtual terminal from below (f_2 in Figure 2), and so on. Since there are w pieces of free places for second virtual terminals on the line g for the w pieces of first virtual terminals $(f_1, f_2, \ldots, f_w)$, there will always be a free place for the second virtual terminal (either to the right or to the left from s_i) on the line g.

It is easy to verify that the wire segments belonging to different nets cannot cross each other, because the wire segments of type a are only 1-unit long, so only the wire segments of type b and c belonging to different nets can cross each other. Assume that a wire segment of type b belonging to the first virtual terminal f_x crosses a wire segment of type c belonging to f_y (in this case, f_y is situated higher). But this means, that we did not choose the first free place for the second virtual terminal s_x when we selected a place on the line g for f_x, since the place for the second virtual terminal s_y was closer to f_x, and it was free too (because we assigned a place for the second virtual terminal s_x earlier than for s_y, since f_x is lower than f_y).

In Figure 2, the dashed lines are situated on the $x = a + 1$ n-plane, and the second virtual terminals are on the $x = a$ n-plane. The connection between the dashed lines and the second virtual terminals could easily be solved by a 1-unit n-wire segment, but it is not worth doing this: for the algorithm used for solving Phase III it is better if the second virtual terminals are actually on the $x = a + 1$ n-plane.

3.4 Phase III

So far, we solved the routing between the terminals on the top grid and the second virtual terminals. It is easy to see that the problem of connecting the second virtual terminals s_i with the bottom terminals b_i is essentially a 2-layer channel routing problem since all the wire segments until now were above the height $z = h_0$ (the two layers correspond to the $x = a$ and $x = a + 1$ n-planes).

This channel routing problem can be solved using Theorem 1. Furthermore, since this channel routing problem is bipartite (that is, each net contains one terminal on the upper, and one on the lower boundary), the required width is at most the length of the channel routing problem, which corresponds to the width w of the 3DCRP. Therefore $h_0 \leq w \leq n$.

Now we have connected all the virtual terminals s_i with the terminals b_i on the bottom grid.

This means that we routed each net. For the wiring between the terminals on the top grid and the second virtual terminals we needed a height of $2n$, and for the wiring between the second virtual terminals and the terminals on the bottom grid, we needed a height of $h_0 = n$. So all in all, the problem can be solved with a height of $2n + n = 3n$.

4 Algorithmic aspects

In the general case of multiterminal nets, we have to solve the SALRP, too. In [10], the presented algorithm works in $O(t \cdot (w + n))$ time. In the solution of the bipartite 3DCRP, Phase I and Phase II can be performed in linear time in the size A of the input (where $A = w \cdot n$ is the area of the planar grid containing the terminals), provided that an edge-colouring of the graph G in section 3.2 is given.

So there are only two real tasks: the edge-colouring of the graph G (so how to assign the access lines to the nets), and the algorithm of Theorem 1 (phase III in the routing). By Theorem 1, the second task can be performed in $O(w)$ time for each row of terminals on the bottom grid. Since we have to solve n pieces of 2-layer channel routing problems, this requires $O(w \cdot n) = O(A)$ time.

For edge-colouring the graph G, there exists an algorithm to edge-colour a bipartite graph with v vertices and m edges in $O(v \cdot m)$ time, so in case of G, this means $O(t \cdot (w + n))$. So all in all, our algorithm works in $O(t \cdot (w + n))$ time, where t is the number of nets. Therefore, if $w = \Theta(n)$ is assumed, the running time of the algorithm is $O(A^{\frac{3}{2}})$, where A is the size of the input.

References

[1] Aggarwal, A., M. Klawe, D. Lichtenstein, N. Linial and A. Wigderson, A lower bound on the area of permutation layouts, *Algorithmica* (1991) **6**, 241-255.

[2] Aggarwal, A., J. Kleinberg and D. P. Williamson, Node-disjoint paths on the mesh and a new trade-off in VLSI layout, *SIAM J. Computing* (2000) **29**, 1321-1333.

[3] Cutler, M. and Y. Shiloach, Permutation layout, *Networks* (1978) **8**, 253-278.

[4] Enbody, R.J., G. Lynn and K. H. Tan, Routing the 3-D chip, *Proc. 28th ACM/IEEE Design Automation Conf.* (1991) 132-137 .

[5] Games, R. A., Optimal book embeddings of the FFT, Benes, and barrel shifter networks, *Algorithmica* (1986) **1**, 233-250.

[6] Leighton, T., Complexity issues in VLSI: Optimal layouts for the shuffle-exchange graph and other networks, *The MIT Press, Cambridge, MA.* (1983)

[7] Leighton, T. and A. L. Rosenberg, Three-dimensional circuit layouts, *SIAM J. Computing* (1986) **15**,793-813 .

[8] Leighton, T., S. Rao and A. Srinivasan, New algorithmic aspects of the Local Lemma with applications to routing and partitioning, *Proc. Tenth Annual ACM-SIAM Symp. on Discrete Algorithms* ACM/SIAM, New York and Philadelphia (1999) 643-652.

[9] Recski A. and F. Strzyzewski, Vertex-disjoint channel routing on two layers, *Integer programming and combinatorial optimization*, 397-405, Ravi Kannan and W.R. Pulleyblank, eds., University of Waterloo Press (1990).

[10] Recski, A. and D. Szeszlér, Routing vertex disjoint Steiner-trees in a cubic grid — an application in VLSI, Submitted

[11] Recski, A. and D. Szeszlér, 3-dimensional single active layer routing, *Discrete and Computational Geometry*, 318-329, Lecture Notes in Computer Science **2098**, Springer, Berlin (2001).

[12] Rosenberg, A. L., Three-dimensional VLSI: a case study, *J. ACM.* (1983) **30**, 397-416.

[13] Shirakawa, I., Some comments on permutation layout, *Networks* (1980) **10**, 179-182.