Journal of Interconnection Networks
Vol. 8, No. 2 (2007) 101–118

World Scientific
www.worldscientific.com

LAYOUT OF AN ARBITRARY PERMUTATION IN A MINIMAL RIGHT TRIANGLE AREA

MARIA ARTISHCHEV-ZAPOLOTSKY
Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel
zmaria@cs.technion.ac.il

YEFIM DINITZ
Department of Computer Science, Ben-Gurion University, Be'er-Sheva 84105, Israel
dinitz@cs.bgu.ac.il

SHIMON EVEN*
Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel

VLADIMIR YANOVSKY
Computer Science Department, Technion – Israel Institute of Technology, Haifa 32000, Israel
volodyan@cs.technion.ac.il

Received 31 July 2005
Accepted 21 May 2007

In VLSI layout of interconnection networks, routing two-point nets in some restricted area is one of the central operations. The main aim is usually minimization of the layout area, while reducing the number of wire bends is also very useful. In this paper, we consider connecting a set of N inputs on a line to a set of N outputs on a perpendicular line inside a right triangle shaped area, where the order of the outputs is a given permutation of the order of the corresponding inputs. Such triangles were used, for example, by Dinitz, Even, and Artishchev-Zapolotsky for an optimal layout of the Butterfly network. That layout was of a particular permutation, while here we solve the problem for an *arbitrary permutation* case. We show two layouts in an optimal area of $\frac{1}{2}N^2 + o(N^2)$, with $O(N)$ bends each. We prove that the first layout requires the absolutely minimum area and yields the irreducible number of bends, while containing knock-knees. The second one eliminates knock-knees, still keeping a constant number, 7, of bends per connection. As well, we prove a lower bound of $3N - o(N)$ for the number of bends in the worst case layout in an optimal area of $\frac{1}{2}N^2 + o(N^2)$.

Keywords: Triangular layout; minimal area; minimal bend number.

*This research was supported by the Fund for the Promotion of Research at the Technion.

1. Introduction

1.1. *Problem Background*

The construction of efficient VLSI layouts for interconnection networks is important, since it improves performance of the resulting parallel architecture, both by reducing its cost (fewer chips, boards, and assemblies) and by lowering various performance impacts, such as signal propagation delay, drive power, and fraction of data transfers to off-chip destinations. Routing two-point nets in some restricted area is one of the central operations in the above problem class. A vast amount of work has been done on routing in a channel (usually, a rectangular area), where the net terminals are placed on the channel boundary (see *e.g.* [1, 4, 5]). The main aim of these works is minimization of the layout area, while other aspects (as a number of bends, a wire length, etc.) are also taken into account as secondary, but still important, parameters for optimization.

Although the most popular channel shape is a rectangle, different forms are also of substantial interest. For example, triangular, staircase, and L-shaped channels are discussed in [2]. There, this is done in the 2-layer Manhattan-diagonal layout model.

In this paper, we consider a right triangle shaped channel under the Thompson layout model [6]. The net terminals are placed on the legs (non-hypotenuse sides) of the right triangle: inputs on one side and outputs on the second one. The order of the outputs is a certain permutation of the order of the corresponding inputs. Our aim is to find such a layout for which the area of the encompassing triangle is minimum.

Such a triangle is used, for example, in [3], for an optimal layout of the Butterfly network – see Figure 1(a). The optimality in that work is obtained by using the triangular layout – as shown in Figure 1(b) – that allows to reduce the whole layout area from $N^2 + o(N^2)$ to $\frac{1}{2}N^2 + o(N^2)$. However, that layout is of a particular permutation – the transpose bijection. In contrast, our aim is a universal solution for an *arbitrary* permutation.

Another natural application of wiring permutations in triangles arises in VLSI technology as follows: Assume there are four subsystems, A, B, C and D, arranged as in Figure 2, where terminals of A must be wired to terminals of B, and terminals of C must be wired to terminals of D, while all these terminals are on the boundary of a square wiring area W. By using two wirings of permutations in two triangles, the desired wiring can be implemented efficiently.

1.2. *Our Main Results*

We present here a universal solution, which works for every permutation. In addition to the generation of a layout which is essentially of minimum area, our solution aims to reduce the total number of bends in the connecting paths. The motivation of reducing the number of bends arises from the fact that in chip manufacturing

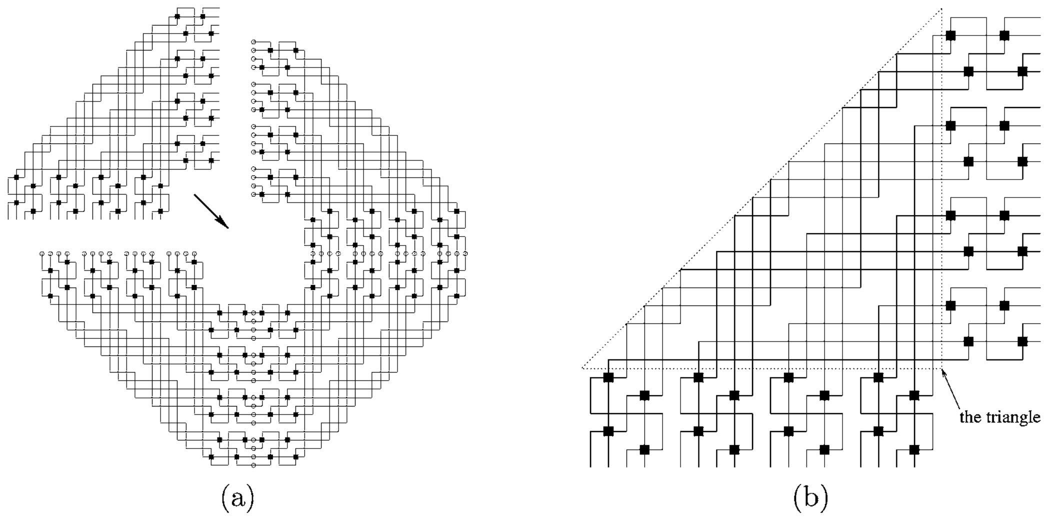

Fig. 1. The optimal Butterfly network layout: (a) The whole layout for $N = 32$, (b) A part of the layout in (a) as an example of the triangular layout of a permutation.

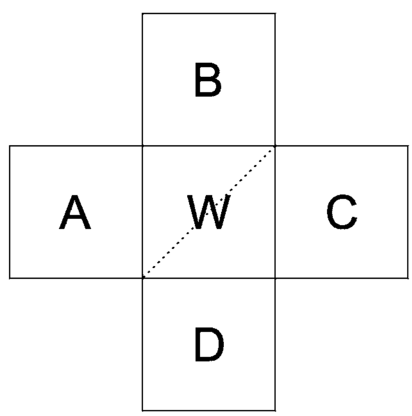

Fig. 2. A VLSI layout case, which uses two wirings of permutations in triangles.

each bend requires a transition hole. A high number of such holes may increase the layout area and the cost of the chip.

We show two solutions:

(1) A layout with knock-knees in a *minimal area* of $\frac{1}{2}(N+1)^2$, for an arbitrary permutation P. The layout contains $3N - 2c(P)$ bends, where $c(P)$ is the number of cycles in P. We prove that the above *bend number* is the *irreducible minimum* (given the minimal area), for *any* permutation. Hence, the layout is optimal w.r.t.[a] area and bend number.

(2) A layout *without knock-knees* in an area of $\frac{1}{2}N^2 + o(N^2)$. We prove that it

[a]with respect to

contains less than $7N$ bends in the worst case, *i.e.* there is still a constant number (in average) of bends per connection. In what concerns the optimality of the bend number, we prove the lower bound of $3N - o(N)$ (given the area of $\frac{1}{2}N^2 + o(N^2)$), for the worst case over all permutations.

The suggested knock-knee elimination method is a general one, and may be applied to any monotonic layout of a permutation in a right triangle, while keeping the area essentially the same and increasing the number of bends quite modestly.

2. Problem Description

Given a set $\mathcal{N} = \{1, 2, \ldots, N\}$ and a permutation $P : \mathcal{N} \mapsto \mathcal{N}$, we want to find a *layout* of P on the square grid, for which the following properties hold (see Figure 3(b), for illustration):

- There is a set $X = \{x_i \mid i \in \mathcal{N}\}$ of vertices positioned in the natural order, ascending from left to right, on distinct integral points of the x-axis. There may be gaps (vacant integral points in between the vertices). Let us denote the interval which includes all the vertices on the x-axis by Ξ.
- There is a set $Y = \{y_i \mid i \in \mathcal{N}\}$ of vertices positioned in the natural order, ascending from bottom up, on distinct integral points of the y-axis. Again, there may be gaps. Let us denote the interval which includes all the vertices on the y-axis by Υ.
- Ξ is located on the l.h.s.[b] and below Υ, and the two intervals are disjoint.
- For every $i \in \mathcal{N}$, there is a grid path, called a *wire*, connecting vertex x_i with vertex $y_{P(i)}$. No edge on the axes is used in these wires. No two wires share a grid edge.
- If two wires intersect at a grid point, they cross each other or both of them bend at this point. The latter case is called a *knock-knee.*
- The layout area is restricted by a right angled triangle such that Ξ is on its horizontal leg and Υ is on its vertical leg, *i.e.* all points of the wires are inside the triangle or on its boundary.

Our aim is to find such a layout for which the area of the encompassing triangle is minimum.

3. Layout with Knock-Knees

Given a permutation $P : \mathcal{N} \mapsto \mathcal{N}$, a *cycle* of *length* k is an ordered subset of $\mathcal{N}$, $(i_1, i_2, \ldots, i_k)$, such that for every $1 \le j < k$, $P(i_j) = i_{j+1}$, and $P(i_k) = i_1$. Clearly, the cycles are disjoint and the union of all of them is $\mathcal{N}$. Let us denote the number of cycles of P by $c(P)$. If $k = 1$ then the cycle is called a *singleton.*

[b]Left hand side; similarly, r.h.s. is for right hand side.

In this section we allow knock-knees. We describe a simple algorithm for a layout of a given P in a right angled isosceles triangle of area $\frac{1}{2}(N+1)^2$. The number of bends is $3N - 2c(P)$, so that N bends occur at the hypotenuse, and there are $(N - c(P))$ knock-knees. The area of the encompassing triangle is minimum, and we show that the number of bends and knock-knees in our layout is minimum for any layout within that triangle.

3.1. *Layout Algorithm*

We start by putting $x_1, x_2, \ldots, x_N$, in this order, on the horizontal side, from left to right, and $y_1, y_2, \ldots, y_N$ on the vertical side, from bottom up, and there are no gaps in either case. Since no edges of the axes are allowed to be used by the wires, the (possibly minimal) length of each leg of the triangle is $N+1$, and consequently its area is $\frac{1}{2}(N+1)^2$ (also the minimum).

We wire each cycle $(i_1, i_2, \ldots, i_k)$ separately. We use only the columns where $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ are located and the rows where $y_{i_1}, y_{i_2}, \ldots, y_{i_k}$ are located, so separate wiring is legal. The wiring of a cycle is done by recursion on the cycle length. The recursive step is described in the following two paragraphs.

If the cycle is a singleton, (i_1), then we start at vertex x_{i_1}, go up to the hypotenuse, turn right, and go all the way to y_{i_1}.

If $k > 1$, let i_a be the least integer in the cycle. Start a wire at $x_{i_{a-1}}$, go up to the row of y_{i_a} (the lowest output in the cycle), turn right and continue all the way to y_{i_a}. Since $i_a < i_{a-1}$, this wire does not reach the hypotenuse. Next, start a wire at x_{i_a}, go up to the hypotenuse (this will occur at the row of y_{i_a}), turn right and go to the column of $x_{i_{a-1}}$. At this point, turn up, creating a knock-knee, and stop the wire, temporarily. For every $j \neq a, (a-1)$ start a wire at x_{i_j} and go up, crossing the horizontal wire at the row of y_{i_a}, and stop temporarily. Looking at the $k-1$ open ends, sticking up from the row of y_{i_a}, and all their destinations, we are left with the job of wiring the same cycle, with i_a omitted, in a smaller right angled isosceles triangle whose legs are of length $N + 1 - i_a$.

Consider the following example:

i	1	2	3	4	5	6	7	8	9
$P(i)$	2	7	6	8	4	3	9	1	5

This permutation has two cycles: $(1, 2, 7, 9, 5, 4, 8)$ and $(3, 6)$. The wiring after the first recursive step, applied to $(1, 2, 7, 9, 5, 4, 8)$, is shown in Figure 3(a), and the complete wiring is shown in Figure 3(b).

In each step of the recursion, a bend is created on the hypotenuse. In addition, in each step of the recursion, except the last of the cycle, one knock-knee is created. Thus, in the wiring of a cycle of length k, the number of bends on the hypotenuse is k and the number of knock-knees is $(k-1)$. Summing these values over all $c(P)$ cycles yields that the total number of bends on the hypotenuse is N and the total number

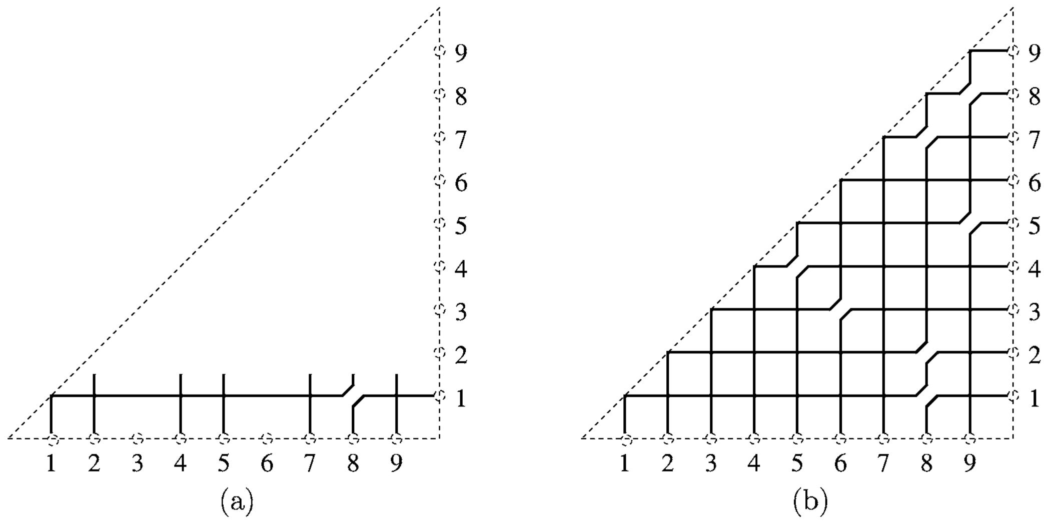

Fig. 3. Wiring of the permutation from the example: (a) wiring the cycle $(1,2,7,9,5,4,8)$ after the first recursive step; (b) the complete wiring.

of knock-knees is $N - c(P)$. Thus, the total number of bends is $N + 2(N - c(P)) = 3N - 2c(P)$.

Proposition 1. *The above algorithm works for any given permutation in a right isosceles triangle of area* $\frac{1}{2}(N+1)^2$. *The area of the encompassing triangle is irreducible minimum. The number of bends is* $3N - 2c(P)$.

In the next subsection, we show that the number of bends is irreducible minimum too (given the minimal area).

3.2. *Optimality of the Number of Bends*

Before we prove the optimality of the layout our algorithm produces, let us make several simple observations:

Lemma 2. *In any wiring of any permutation* $P : \mathcal{N} \mapsto \mathcal{N}$, *in a right angled isosceles triangle whose legs are of length* $N+1$, *all grid-edges inside the triangle are used in the wiring.*

Proof. Any wiring of a connecting path from x_i to y_j must have at least $N - i + 1$ horizontal grid edges in order to reach the y-axis. Thus, the entire wiring of P uses at least $\frac{1}{2}N(N+1)$ horizontal grid edges. However, this is the total number of horizontal grid edges inside the triangle. Hence, every horizontal grid edge is used.

The argument for the vertical grid edges is similar. □

A wire from x_i to y_j is said to be *monotonic* if it goes up and/or right (perhaps several times), but it never goes down or to the left. The number of edges in each such path is, therefore, $N - i + 1 + j$.

Corollary 3. *In a wiring of a permutation, as in Lemma 2, the following three facts hold:*

- *All wires are monotonic.*
- *Every grid-point on the hypotenuse, except its two ends, is used by some wire to bend.*
- *If an internal grid-point is used by any wire to bend, then it is a knock-knee.*

Now, let us prove that the bend number obtained by the algorithm in Section 3.1 is the irreducible minimum.

Theorem 4. In any layout of a permutation $P : \mathcal{N} \mapsto \mathcal{N}$ in a right triangle whose legs are of length $N + 1$, the number of knock-knees is at least $N - c(P)$.

Proof. The proof is by induction on N. For $N = 1$ the theorem is trivial.

Suppose the theorem holds for $N-1$ terminal pairs and any $c(P)$. For N terminal pairs, consider the routing on the lowest row, R, *i.e.* that of y_1.

Let $q = P^{-1}(1)$. By the monotonicity (see Corollary 3), the wire from x_q to y_1 goes up one grid edge and then to the right, ending in y_1. There are two cases to be considered.

If $q = 1$, *i.e.* (1) is a singleton, the wiring on R has no knock-knees. The wires from all other terminals, $x_2,\ x_3, \ldots, x_N$ start with at least two grid edges upward, crossing R, and must be wired to their destinations within the remaining right angled isosceles triangle whose sides are of length N. The permutation $P' : \{2,\ 3, \ldots, N\} \mapsto \{2,\ 3, \ldots, N\}$, which remains to be wired, consists of the same cycles as P, with the exception of (1). By the inductive hypothesis, its layout has at least $(N-1) - (c(P) - 1) = N - c(P)$ knock-knees, proving the theorem.

Thus, assume $q > 1$. In this case, there is a knock-knee in R, at the column of x_q. There are no knock-knees in R on the r.h.s. of x_q, but there may be some on its l.h.s. Assume that there are altogether $t \geq 1$ knock-knees on R. Let P' denote the permutation to be wired above R. We make the following claim:

Claim 5. *The number of cycles $c(P')$ satisfies $c(P') \leq c(P) + t - 1$.*

Once Claim 5 is proven, the theorem follows: The total number of knock-knees is at least $t + (N-1) - c(P')$: t on R and at least $(N-1) - c(P')$ above R, by the induction assumption. By the claim,

$$t + (N-1) - c(P') \geq t + (N-1) - (c(P) + t - 1) = N - c(P)\ .$$

Proof of Claim 5. The proof of the claim is by induction on t. If $t = 1$, every cycle of P that does not include 1, remains intact in P'. The cycle C, which includes 1 and q is affected. In this cycle, 1 is dropped, $P'(q) = P(1)$ and for any other i, $P'(i) = P(i)$. Thus, $c(P') = c(P)$, satisfying the claim.

Assume $t > 1$ and that for $t-1$ the claim holds. Let a be the least integer for which there is a knock-knee on R at the column of x_a; call the grid point of this knock-knee, A. By our assumption, $a \neq q$.

Let $P_0 : \mathcal{N} \mapsto \mathcal{N}$ be the permutation defined by the wiring of P, if A is changed to be a cross point, instead of a knock-knee. It follows that $P_0(a) = P(1)$, $P_0(1) = P(a)$, while for every $j \neq 1, a$, $P_0(j) = P(j)$.

If 1 and a are not in the same cycle of P then $c(P_0)$ satisfies $c(P_0) = c(P) - 1$, since the two cycles in P, the one in which 1 is and the one in which a is, are now changed into one cycle, and no other cycle is affected.

If 1 and a are in the same cycle C of P then in P_0 the cycle changes into two cycles, but again, no other cycle is affected. Therefore, in this case, $c(P_0) = c(P)+1$. Thus,

$$c(P_0) \leq c(P) + 1 \tag{3.2}$$

always holds.

Both in the case of P and P_0, the permutation $P' : \{2, 3, \ldots, N\} \mapsto \{2, 3, \ldots, N\}$, wired above R, is the same. Since P_0 has $t-1$ knock-knees on R, by the inductive hypothesis (of the claim), the number of cycles of P', $c(P')$, satisfies

$$c(P') \leq c(P_0) + (t-1) - 1 \; .$$

By Equation 3.2,

$$c(P') \leq c(P) + 1 + t - 1 - 1 = c(P) + t - 1.$$ □

Theorem 6. In any layout of a permutation $P : \mathcal{N} \mapsto \mathcal{N}$ in a right triangle whose legs are of length $N+1$, the number of bends is at least $3N - 2c(P)$.

Proof. According to Corollary 3, there are exactly N bends on the hypotenuse.

Each knock-knee adds two bends to the layout. According to Theorem 4, the number of knock-knees is at least $N - c(P)$, therefore the number of the bends created by the knock-knees is at least $2(N - c(P))$. Thus, the total bend number is $3N - 2c(P)$. □

By Theorem 6 and Proposition 1, the algorithm of Section 3.1 provides the irreducible minimum.

4. Knock-Knee Elimination

In this section, we show how all knock-knees of *any monotonic layout*, in a right triangle, can be removed. This removal increases the area, but we show that the area remains essentially the same. In the case when the original layout is of area $\frac{1}{2}N^2 + o(N^2)$, as that of the previous section, the area of the fixed layout remains the same, except that the $o(N^2)$ expression hides a higher value.

Suppose we are given a monotonic layout L of a permutation $P : \mathcal{N} \mapsto \mathcal{N}$ in a right triangle, and L contains knock-knees. Our aim is to show how to rewire the connections, so that there will be no knock-knees, preserving P and keeping the area essentially the same. Our approach is to divide the triangle into blocks, and to define a rewiring, which keeps the macro-structure. That is the sequence of blocks passed by each wire will be the same, but wires change their way through each block, so that to avoid creating knock-knees. The arising "local" non-exactness at outputs is fixed by adding small *channel routers* (blocks performing channel routing).

In the rest of the section, we first consider the layout produced in Section 3.1. In Section 4.1, we describe the knock-knee elimination algorithm and study its properties. Since our method enlarges the layout area, we state and prove a lower bound on the bend number for the case of a "non-tight" area, in Section 4.2. Finally, in Section 4.3, our knock-knee elimination method is generalized to any monotonic layout in a right triangle.

4.1. *Rewiring in Blocks*

In Sections 4.1-4.2, we deal with layout L produced by the algorithm of Section 3.1. For simplicity of presentation, we first assume that $N = n^2$, for some natural n.

First, divide the input vector into n *small vectors*, each consisting of n input terminals. Next, introduce a gap between every two small vectors. This introduces $n - 1$ new columns; none of its grid edges is used, but the horizontal wires crossing such a column are extended by one unit. Similarly, introduce gaps in the output vector, and their new rows.

What we have now is a division of the layout into *blocks*. Figure 4 gives an example of such a division, for $N = 16$. There are n triangular blocks on the diagonal, each has n inputs on its south side and n outputs on its east side. The remaining $\frac{1}{2}n \cdot (n - 1)$ blocks are square shaped, with "local" inputs at the south and west sides, and "local" outputs at the north and east sides. For each block, we define a rewiring, keeping the *sets* of wires coming from south and from west, and those leaving to north and to east, as they are in the original layout.

Let us consider a particular block, parametrized by the set of incoming wires and just outcoming *directions*, for each one of them. Each wire comes at its particular point on the south or west boundary (or south only, for triangular blocks) and aims to the east or north (or east only, for triangles). First, we show a method of wiring in a block, which produces a knock-knee free layout obeying these requirements. After that, we show a block processing sequence, which provides the parametrization as above, for each block processed. Finally, we adjust the placement of outputs.

4.1.1. *Rewiring in a Triangular Block*

The wiring of each triangular block is done simply by going up from each input to the diagonal, bending right and going to the east terminal of the block (as in Figure 4). This may not preserve the original (as in L) order of the outputs on the

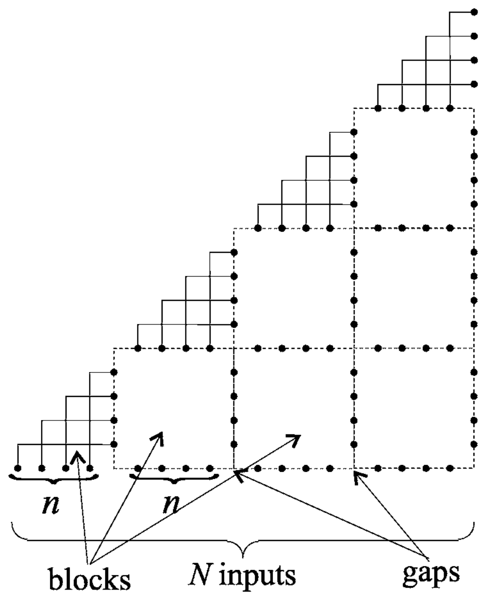

Fig. 4. The division into blocks.

east boundary of this block, thus defining a new order of the inputs on the west boundary of the next block of that horizontal layer. However, this ensures that all the wires supposed to enter the next block from the west do so, despite a different order (w.r.t. L).

4.1.2. *Rewiring in a Square Block*

The wiring of the connections through a block is described below. For illustration, see in Figure 5 wiring of an example, where n equals 6, $W1$, $W2$, $W5$, $W6$ must go north and $S1$, $S3$, $S4$, $S6$ must go east.

(1) *Straight wires.* Every wire which enters the block from the west, say at terminal Wi, and is supposed to leave from the east, goes straight across to terminal Ei (wires $W3 - E3$, $W4 - E4$ in Figure 5). Similarly, every wire which enters from the south, say at terminal Sj, and is supposed to leave from the north, goes straight up to terminal Nj (wires $S2 - N2$, $S5 - N5$ in Figure 5).

(2) *Bending wires.* If there are wires incoming from the west and supposed to leave from the north, let a denote the number of these wires. Clearly, this is also the number of wires entering the block on the south side to go east. The number of currently unused terminals on each block side is also a.

- *The detour wire.* Let Wp_1, $Wp_2, \ldots,$ Wp_a be the unused terminals on the west side, in ascending order, and let Nq_1, $Nq_2, \ldots,$ Nq_a be the unused terminals on the north side, in ascending order. The connection from Wp_1 bends up on the west boundary of the block, goes all the way up to

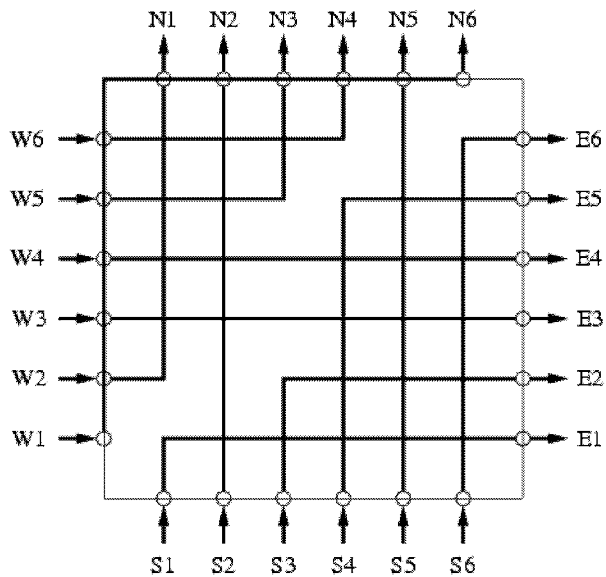

Fig. 5. The wiring of the example square block.

the north-west corner of the block, bends right and goes along the north boundary up to Nq_a (wire $W1 - N6$ in Figure 5).

- *West-north bending wires.* For each $2 \leq i \leq a$, the wiring from Wp_i goes horizontally until it reaches the column of terminal Nq_{i-1} (still unused). It bends up and goes all the way to Nq_{i-1} (wires $W2 - N1$, $W5 - N3$ and $W6 - N4$ in Figure 5).
- *South-east bending wires.* Let $Sq_1, Sq_2, \ldots, Sq_a$ be the terminals on the south side, for which the connections (to the east side) have not yet been wired, and $Ep_1, Ep_2, \ldots, Ep_a$ be the terminals on the east side, which are yet unused, both in ascending order. For each $1 \leq i \leq a$, we wire the connection from Sq_i by going up p_i units, bending right and going straight to Ep_i (wires $S1 - E1$, $S3 - E2$, $S4 - E5$ and $S6 - E6$ in Figure 5).

Lemma 7. *For the above wiring in a square block, the following facts hold:*

(1) There are no overlaps and knock-knees.
(2) The number of bends is bounded from above by $2a + 2$*, where* $2a$ *is the number of all bending (west-north or south-east) connections.*

Proof.

(1) Wiring of straight and west-north connections does not cause any wire contentions, since the row/column used by each such a connection is not used neither by other straight connections, nor by bending connections.

When wiring south-east connections, each wire from Sq_i goes up p_i units, arriving at the row of Wp_i. It does not reach the vertical section of the connection to Nq_i, since the latter starts at the row of Wp_{i+1}, which is higher than the row of Wp_i. Thus, in the columns, no overlaps and knock-knees are created.

The situation with the rows is symmetric: the wire from Sq_i uses row p_i starting from column q_i, while the wire from Wp_i goes just until the column of Nq_{i-1}. Since $q_{i-1} < q_i$, this appears also to be legal and knock-knee free.

(2) There is one bend for each bending connection, except for the detour wire, which requires two more bends. So, in the case of $2a$ bending wires, the number of bends is $2a + 2$. □

4.1.3. *Rewiring in the Entire Triangle*

We begin rewiring by processing the lowest block row, starting from the leftmost block and proceeding from left to right. For all blocks, the incoming points for wires going from south are pregiven by the original layout. For the first, triangular block, this information provides its full parametrization. For any other block, the placement of wires going from the west is defined by wiring of the previous block. The outcoming direction, north or west, for any wire, is pregiven by the original layout. Since this parametrization is full, the blocks may be processed as in Sections 4.1.1 and 4.1.2. Once a layer is done, we proceed with the layer on top of it, in the same fashion. The wires outcoming from the blocks of the previous layer to the north are those incoming to the blocks above them from the south. Thus, each layer has a similar initial information, sufficient for parametrizing its blocks.

The order of outputs, in each block, is not controlled by our approach. Therefore, the resulting permutation of the wires may be not consistent with P. However, the wire set of each small output vector is preserved. We rewire now the outputs in each small vector to be according to P. Connecting the wires to their exact destinations is done using Pinter's algorithm for channel routing [5]. The width of such a channel is bounded by $1.5n$. The resulting layout looks like in Figure 6.

4.1.4. *Knock-Knee Elimination for Non-Quadratic N*

When $\sqrt{N}$ is non-integer, let n be the least number such that $n^2 > N$. Let $d = (N \bmod n)$ and $D = N - d$. Then, we suggest the following division into blocks: First D inputs (in the natural order) are divided into small vectors of length n each, and if $d > 0$, the additional, last small vector contains d vertices; it makes $n - 1$ or n vectors. The same is done for outputs. For illustration see Figure 7.

Now, we have a number of triangular, square and rectangular blocks. All the triangles are isosceles (just vary in size), therefore the rewiring algorithm for them does not change.

Rectangular blocks may seem to cause a problem, but actually their rewiring goes along the same lines, as for square ones. Notice, that if there are a wires that enter some rectangular block from the west and go to the north, then exactly a outputs on the east side are left for wires going from south inputs. Therefore, all the

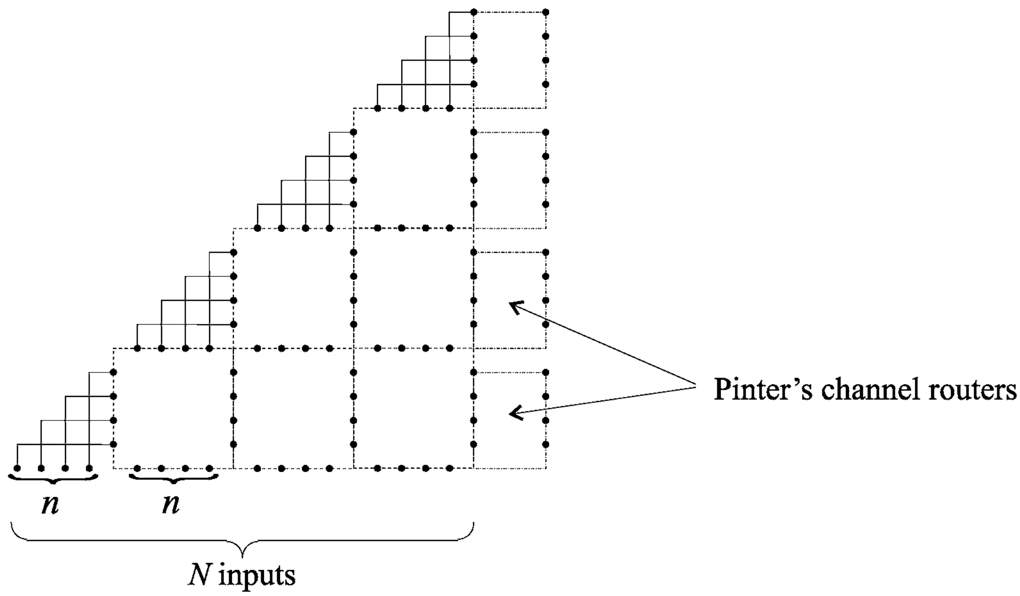

Fig. 6. The whole layout scheme based on the division into blocks.

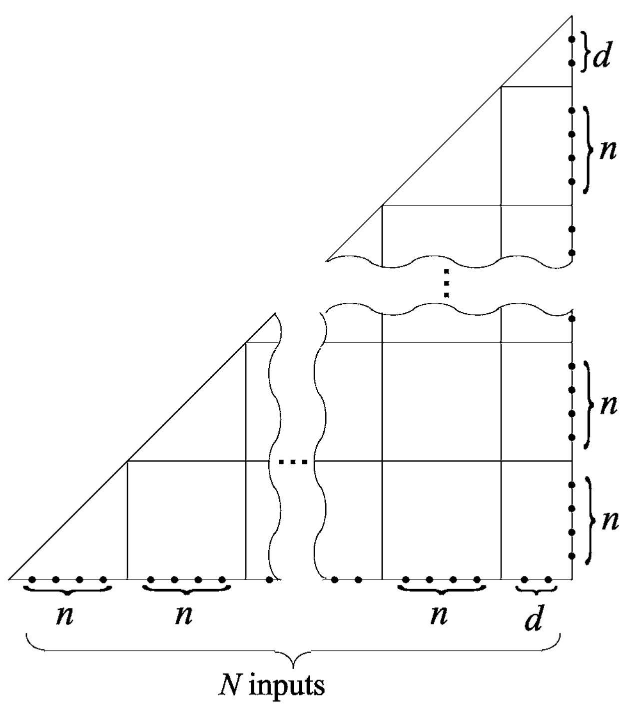

Fig. 7. Non-equal division into blocks.

steps of the rewiring algorithm for square blocks will do well for the rectangles. Just the number of horizontal straight wires will be greater than the vertical ones. Thus, the above rewiring algorithm for knock-knee elimination works for non-quadratic values of N as well.

4.1.5. *Properties of the rewired layout*

Let us examine the area and bend number of the new layout. From here on, we refer as "rectangular" both rectangular and square blocks.

Theorem 8. The suggested layout has the following properties:

(1) It is legal w.r.t. the Thompson model and has no knock-knees.
(2) The layout area is $\frac{1}{2}N^2 + o(N^2)$.
(3) The number of bends is less than $7N$.

Proof.

(1) By Lemma 7, there are no overlaps and knock-knees inside the blocks. In rewiring of a rectangular block, only the north and west boundaries of each block may be used by a wire (the detour one). Therefore, when two blocks are juxtaposed, having a common side, no overlaps and knock-knees are created, since only one of them uses the shared side. Thus, the entire layout does not contain overlaps and knock-knees.
(2) Obviously, the new layout is also encompassed by a right isosceles triangle. The length of its leg is bounded by $(N+1)+(n-1)+1.5n$, where $n < \sqrt{N}+1 = o(N)$, $(N+1)$ is the leg length of the original triangle (described in Section 3), at most $(n-1)$ is added by the gaps between the blocks, and $1.5n$ is the worst case width of Pinter's channel routers. Therefore, the overall area is $\frac{1}{2}N^2 + o(N^2)$.
(3) Let us compare the number of bends in the new layout with that of the optimal layout suggested in Section 3. For convenience, let us imaginarily divide also the optimal layout into blocks in the same fashion, as this is made in Section 4.1. We consider 3 types of blocks:

- In the knock-knee free layout, each triangular block with l inputs contains exactly l bends. The corresponding triangular block in the optimal layout contains at least l bends (there may be also knock-knees inside). Thus, rewiring of triangular blocks does not add bends.
- Each rectangular block in the new layout preserves the number of connections that change the direction inside the block. Therefore, if the rewired block contains $2a + 2$ bends (see Lemma 7), then the corresponding rectangular block in the optimal layout contains at least $2a$ bends (there may be wires that bend several times inside the block). So, rewiring of any rectangular block adds at most 2 bends. Let n' denote the number of small vectors subdividing inputs ($n - 1$ or n). The overall number of rectangular blocks is $\frac{n'(n'-1)}{2}$. Therefore, the number of additional bends is at most

$n'(n'-1) < N$. [c]

- At last, the Pinter channel routers do not appear in the optimal layout, so their bends make a pure addition to the optimal number of bends. By [5], this is $3n$ per block (in the worst case), or at most $3N$ altogether.

Hence, the total number of additional bends is less than $N + 3N = 4N$ in the worst case. Therefore, number of bends in the new layout is less than $3N - 2c(P) + 4N < 7N$. □

So, the number of bends in the new layout is less than 7 per connection, in average.

4.2. *General Lower Bound on the Bend Number*

Theorem 6 states the lower bound of $3N - 2c(P)$ on the number of bends, in the case of *absolutely minimal area*, *i.e.* area that equals exactly $\frac{1}{2}(N+1)^2$; it is valid for any permutaion P. While eliminating knock-knees, we extend the layout area to $\frac{1}{2}N^2 + o(N^2)$, by lengthening the triangle legs by $o(N)$. The following study implies the worst case lower bound of $3N - o(N)$ on the number of bends, for any layout in such an area. Therefore, the number of bends achieved in Section 4.1.3, while eliminating knock-knees, has a bit more than twice the minimum possible bend number for a layout with knock-knees permitted.

Besides, we make our lower bound more precise as related to the number of cycles. Proposition 1 and Theorem 6 hint that the number of needed bends decreases, when the number of cycles increases. For our hard permutation instances, the number of bends depends on the number of cycles in the same form $3N - 2c(P)$, as in those Proposition and Theorem.

Note that if the leg of a right isoceles triangle is at least $2N$, any input may be wired to the corresponding output straightforwardly, with a single bend. Hence, in what follows we do not consider this case.

Theorem 9. For any $N \geq 1$, any $\nu : 0 \leq \nu < N$, and any $c : \nu \leq c \leq N$, there exists a permutation $P : \mathcal{N} \mapsto \mathcal{N}$, with $c(P) = c$, such that any layout of P in a right isoceles triangle, whose legs are of length $N + \nu$, has at least $3N - 2c$ bends. [d]

Proof. For any pair N and ν, consider the right triangle with the legs of length $N+\nu$. Let $C_1, C_2, \ldots, C_{N+\nu-1}$ denote the grid columns inside the triangle, from the

[c] This inequality can be trivially proven by examining two cases:

(a) If $(n-1)^2 < N \leq n^2 - n$, then $n' = n - 1$.
(b) If $n^2 - n < N \leq n^2$, then $n' = n$.

[d] For any $c < \nu$, we can create a permutation with bend number at least $3N - \nu$. It is obtained by constructing a permutation like P_0 in the theorem proof and then exchanging targets for some values greater than $N - \nu$.

left to the right; and let $R_1, R_2, \ldots, R_{N+\nu-1}$ denote the inner grid rows from the bottom up.

Consider any layout in this triangle. Notice that any input-output connection has at least one bend, since it starts with a vertical segment and finishes with a horizontal one. Besides, any connection that starts at column C_i and finishes at row R_j, with $i < j$, has to make at least 2 additional bends in order to keep itself inside the triangle area (otherwise, it would stick out of the hypotenuse). Let A denote the set of such connections. Then, the overall number of bends in the layout is at least $N + 2|A|$.

Let us first ignore the requirement concerning the cycle number, for simplicity. Consider the permutation $P_0 : \mathcal{N} \mapsto \mathcal{N}$, where for every $1 \leq i \leq N$, $P_0(i) = ((i + \nu - 1) \mod N) + 1$. Consider any $q : 1 \leq q \leq N - \nu$. Since $q + \nu - 1 \leq N - 1$, then $P_0(q) = ((q+\nu-1) \mod N)+1 = (q+\nu-1)+1 = q+\nu$. Let C_a be the column where x_q is placed, and R_b be the row of $y_{P_0(q)}$. Since there are $N - q$ terminals on the r.h.s. of x_q, then $a \leq (N + \nu - 1) - (N - q) = q + \nu - 1 < P_0(q)$. On the other hand, b is at least $P_0(q)$. Therefore, $a < b$, and, consequently, the connection from x_q to $y_{P_0(q)}$ belongs to A. Thus, $|A| \geq N - \nu$, and the number of bends is therefore at least $N + 2(N - \nu) = 3N - 2\nu$.

We continue by constructing a bit different permutation P, with exactly c cycles (for illustration see Figure 8):

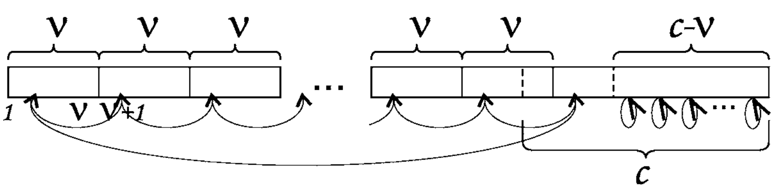

Fig. 8. The cycles in the permutation P.

$$P(i) = \begin{cases} i + \nu & \text{if } 1 \leq i \leq N - c \\ ((i-1) \bmod \nu) + 1 & \text{if } N - c < i \leq N - c + \nu \\ i & \text{if } N - c + \nu < i \leq N \end{cases}$$

It is easy to see that all numbers up to $N - c + \nu$ form exactly ν cycles, each consisting of the set of numbers equal modulo ν. The remaining $c - \nu$ numbers form $c - \nu$ singletons, that is $c(P) = c$, as required. Similarly to the above, for any number q up to $N - c$, the connection from x_q to $y_{P(q)}$ belongs to A, for P. Hence, $|A| \geq N - c$, and the number of bends is therefore at least $N + 2(N - c) = 3N - 2c$.
□

Corollary 10. *For any $N \geq 1$ and any positive $c \leq N$, there exists a permutation $P : \mathcal{N} \mapsto \mathcal{N}$, with c cycles, such that any layout of P in a right isoceles triangle, whose legs are of length $N + c$, has at least $3N - 2c$ bends.*

Corollary 11. *For any $N \geq 1$ and any positive $\nu = o(N)$, there exists a permutation $P : \mathcal{N} \mapsto \mathcal{N}$, such that any layout of P in a right isoceles triangle, whose legs are of length $N + \nu$, has $3N - o(N)$ bends.*

4.3. *Rewiring Any Monotonic Triangular Layout*

Let us consider an arbitrary ready *monotonic* layout with knock-knees, in a right isoceles triangle, possibly containing gaps among its inputs and outputs. Let us apply the above method in order to rewire this layout to make it knock-knee free. The rewiring algorithm is similar to that described earlier. The only difference in the situation is that blocks (both triangular and rectangular) may have unused inputs and outputs.

Obviously, it is not a problem for a triangular block, since each connection inside it is wired completely independently. Therefore, unused inputs just cause corresponding columns and rows to be unused too, and the block becomes sparce.

Let us consider a rectangular block with m unused inputs (at the south and west sides altogether). Naturally, it has also m unused outputs at the north and east sides. Now, just complete the set of input-output pairs for this block by m virtual ones, matching unused inputs with unused outputs in an arbitrary way. For this complete case, we may apply the solution described in Section 4.1, as it is. Finally, erase from the obtained layout the virtual connections to get the sparce block layout.

Theorem 12. For any layout of any permutation $P : \mathcal{N} \mapsto \mathcal{N}$ in a right triangle, whose legs are of length $N_1 + 1$, $N_1 \geq N$, and whose number of bends is B_1, the layout corrected, as in this section, has the following properties*:*

(1) It is legal w.r.t. the Thompson model and has no knock-knees.
(2) The layout area is $\frac{1}{2}{N_1}^2 + o({N_1}^2)$.
(3) The number of bends is less than $B_1 + N_1 + 3N$.

The proof is similar to that of Theorem 8.

Acknowledgment

The authors thank Ami Litman for posing the question about a triangular layout for an arbitrary permutation.

References

1. Bonnie Berger, Martin Brady, Donna Brown, and Tom Leighton. Nearly optimal algorithms and bounds for multilayer channel routing. *J. ACM*, 42(2):500–542, 1995.
2. Sandip Das, Susmita Sur-Kolay, and Bhargab B. Bhattacharya. Routing of L-shaped channels, switchboxes and staircases in manhattan-diagonal model. In *VLSID '98: Proceedings of the Eleventh International Conference on VLSI Design: VLSI for Signal Processing*, pages 65–70. IEEE Computer Society, 1998.

3. Yefim Dinitz, Shimon Even, and Maria Zapolotsky. A compact layout of the butterfly. *J. of Interconnection Networks*, 4(1):53–75, 2003.
4. S. Muthukrishnan, Mike Paterson, Süleyman Cenk Şahinalp, and Torsten Suel. Compact grid layouts of multi-level networks. In *STOC: ACM Symposium on Theory of Computing (STOC)*, pages 455–463, 1999.
5. Ron Y. Pinter. On routing two-point nets across a channel. In *ACM IEEE 19th Design Automation Conference Proceedings*, pages 894–902, Los Alamitos, Ca., USA, June 1982. IEEE Computer Society Press.
6. C. D. Thompson. A complexity theory for VLSI. Technical Report CMU-CS-80-140, Carnegie-Mellon University, 1980.