Polynomial Time Manhattan Routing without Doglegs - a Generalization of Gallai's Algorithm

Endre Boros* András Recski† Tibor Szkaliczki‡ Ferenc Wettl§

Revised: April 1998

Abstract

Gallai's classical result on interval packing can be applied in VLSI routing to find, in linear time, a minimum-width dogleg-free routing in the Manhattan model, provided that all the terminals are on one side of a rectangular [1]. Should the terminals appear on two opposite sides of the rectangular, the corresponding "channel routing problem" is $\mathcal{NP}$-complete [2, 3].

We generalize Gallai's result for the case if the terminals appear on two adjacent sides of the rectangular.

Keywords: VLSI, routing, Manhattan model, polynomial algorithm, complexity
Abbreviated title: Polynomial Time Manhattan Routing without Doglegs

1 Introduction

Let s and w be given positive integers, and let us consider the $(w+1) \times (s+1)$ square planar grid graph $G = (V, E)$, where $V = \{\, (x, y) \mid 0 \le x \le s+1 \text{ and } 0 \le y \le w+1 \,\}$, and $E = \{\, ((a, b), (c, d)) \mid (a, b) \in V,\ (c, d) \in V,\ |a - c| + |b - d| = 1,\ 0 < a + c < 2s + 2 \text{ and } 0 < b + d < 2w + 2 \,\}$. The integers s and w are called the length and width, respectively, of G. The points on the boundary of G, with the exception of its four corners, will be called *terminals*. In particular, the points with coordinates $(x, 0) \in G,\ 0 < x \le s$ will be called *Southern terminals*, while those with $(0, y) \in G$, $0 < y \le w$ will be called *Western terminals*. (Eastern and Northern terminals can be defined analogously.)

*RUTCOR, Rutgers Center for Operations Research Rutgers, The State University of New Jersey 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA. (`boros@rutcor.rutgers.edu`)

†Department of Computer Science and Information Theory, Technical University of Budapest, Budapest, Hungary (`recski@math.bme.hu`)

‡Computer and Automation Research Institute, Hungarian Academy of Sciences, H-1111 Budapest, Lágymányosi u. 11, Hungary (`sztibor@ilab.sztaki.hu`)

§Department of Algebra, Technical University of Budapest, Budapest, Hungary (`wettl@math.bme.hu`)

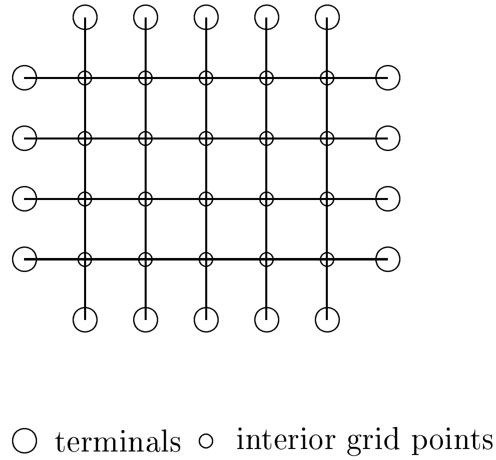

Figure 1: The grid graph G, when $s = 5$ and $w = 4$.

We shall call a set of terminals a *net*, and a collection of pairwise disjoint nets will be called a *routing problem*. (In a routing problem some terminals may belong to no nets.) A *realization* (or *wiring*) of a net is a connected subgraph (usually a tree) of G connecting the terminals belonging to the net, and not containing any other terminals. The restrictions in the definition of E imply that all connections must be realized through the "interior" of G, and no line segments on the boundary can be used as edges. A *solution* (or a *wiring plan*) of a routing problem is a set of *t-disjoint* realizations of all the nets in the problem, where the expression *t-disjoint* refers to some disjointness-type requirement depending on the technology under consideration.

A routing problem is called *single-row routing problem* if every terminal is on one side of G, say e.g. Southern. It is called *channel-routing problem* if the terminals are on two opposite sides, e.g. every terminal is either Northern or Southern. Final, we shall call a routing problem *gamma-routing* if the terminals belong to two adjacent sides, e.g. if every terminal is either Southern or Western. All these are special cases of the *switchbox-routing problems* where the terminals can be at any of the four sides of G. Hence gamma- and channel-routing are two possible generalizations of the single row routing problem while switchbox-routing is their common generalization.

Throughout this paper we shall consider the "Manhattan-model," that is, we suppose 2-layer routing with one layer reserved for the horizontal and one for the vertical line segments. Whenever a horizontal and a vertical line segments are adjacent to $(x, y) \in G$ in a realization of a net, we assume that a *via hole* is punched at (x, y). (Hence changes in wire direction are realized by via holes.) In our paper we shall consider the standard technology in which t-disjointness means edge-disjointness with the additional restriction that two distinct paths may cross each other at a grid point but so called *knock-knees*, i.e. grid points used as turning point by two distinct paths, are prohibited.

For example, if $s = w = 1$ and $\{(0,1),(2,1)\}$ and $\{(1,0),(1,2)\}$ are the two nets in a switchbox-routing problem then there is a solution in the Manhattan-model. However, should the nets be $\{(0,1),(1,0)\}$ and $\{(2,1),(1,2)\}$ then the only edge-disjoint realization would use the point $(1,1)$ as a knock-knee. (Recall that neither the corners nor the other terminals can be used by a net.)

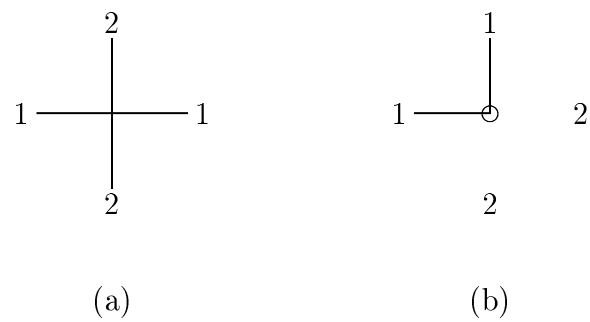

Figure 2: Figure (a) shows a simple routing problem with a feasible solution, while the problem in figure (b) has no feasible solution.

Let us consider a net N consisting of $a = a(N)$ Southern and $b = b(N)$ Western terminals. Then any Manhattan-realization will clearly require at least $a + b - 1$ via-holes if both a and b are positive, and at least $a + b$ via-holes otherwise. Let us call a realization of net N *dogleg-free*, if the number of via holes in it attains the above minimum. For example, in Figure 3 we can see both dogleg-free and non dogleg-free realizations of the same nets.

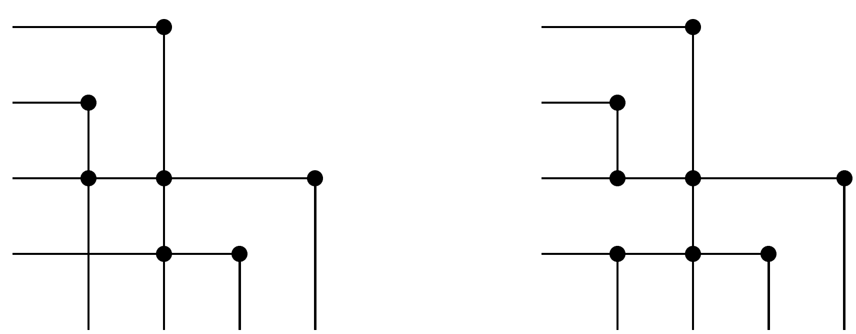

Figure 3: A dogleg-free and a non dogleg-free realization of the same net with 4 Southern and 4 Western terminals.

Recall that the channel routing problem in the Manhattan model, with or without doglegs, is NP-hard [2, 3]. For additional results regarding the complexity of these and related problems see also [4, 5, 6].

In this paper we shall consider gamma-routing problems in which each net N has $\min\{a(N), b(N)\} \leq 1$, and show that Gallai's algorithm can be generalized for this case. Our algorithm runs in polynomial time in the number of grid-points and provides either a dogleg-free solution, or proves that such solution does not exist.

2 Definitions and basic properties

Let us consider a grid-graph $G = (V, E)$ as before with w Western and s Southern terminals, and let us consider a gamma-routing problem for the nets given as $N_r = (W_r, S_r)$, $r = 1, ..., n$, where $W_r \subseteq \{1, 2, ..., w\}$ denotes the set of non-zero coordinates of the western pin positions, and $S_r \subseteq \{1, 2, ..., s\}$ denotes the set of non-zero coordinates of the southern pin positions of the rth net. In other words, points of the form $(0, j)$ for $j \in W_r$ are the Western terminals, while points of the form $(i, 0)$ for $i \in S_r$ are the Southern terminals of N_r.

We shall assume in the sequel that

(i) $W_r \cap W_s = \emptyset$ and $S_r \cap S_s = \emptyset$ for $r \neq s$, and that

(ii) for every net $N_r = (W_r, S_r)$ either $|W_r| \leq 1$ or $|S_r| \leq 1$.

Let us observe first that if $|W_r| = 1$ or $|S_r| = 1$, then N_r has a unique dogleg-free realization, therefore the input of the considered gamma-routing problem consists of the following three types of nets: *West only* nets (i.e. for which $S_r = \emptyset$), *South only* nets (i.e. for which $W_r = \emptyset$), and *uniquely realizable* nets, i.e. those with $|N_r| = 1$ or $|S_r| = 1$.

We can therefore assume without any loss of generality that the uniquely realizable nets are processed first, and let us label the nets such that N_j, $j = 1, \ldots, u$ are the uniquely realizable ones.

Let us define an (i, j)-*realization* of the (West only or South only) net N_r the one which is dogleg-free, and has a via at (i, j), and for which $j = \min W_r$ in case of a West-only net, and $i = \min S_r$ in case of a South-only net (i.e. if N_r is West-only, then it is realized by connecting its pins $(0, j)$, $j \in W_r$ to the vertical line segment between $(i, 0)$ and $(i, \min W_r)$, and placing a via hole at each points of the form (i, k) for $k \in W_r$.)

We shall say that a realization of N_r is in *conflict* with a realization of N_s $(s \neq r)$, if the two realizations are not t-disjoint, i.e. if the two subgraphs would share an edge or a via hole.

Let us denote by F_r the set of gridpoints (i, j) for which there exists an (i, j)-realization of N_r. We shall initialize F_r, $r = u + 1, ..., n$, by including in F_r all grid points for which the corresponding realization of N_r is not in conflict with the (unique and already fixed) realizations of the nets N_j, $j = 1, ..., u$.

Our problem is now to find an assignment $N_r \mapsto (i, j) \in F_r$ such that no two of these realizations are in conflict.

We shall construct such an assignment, by fixing successively the realizations of some of the nets, and then eliminating the conflicting realizations from the sets F_r for the remaining nets N_r. Thus, in any given moment, there will be some nets already realized, and some others, which are still unrealized.

Given such a situation, let us say that the assignment $(i,j) \in F_p$ for an unrealized net N_p is *forbidden*, if there exists another unrealized net N_q for which the (i,j)-realization of N_p is in conflict with all possible realizations from F_q of N_q. We shall also say that $(i,j) \in F_p$ is *infeasible* if it is in conflict with the realization of some of the already realized nets.

Let us finally write $(i,j) \leq (i',j')$ if $i \leq i'$ and $j \leq j'$, and let us say that (i,j) is a *minimal* element of a set S of grid points, if $(i,j) \in S$ and there is no $(i',j') \in S$ for which $(i',j') \leq (i,j)$ and $(i',j') \neq (i,j)$.

3 Proposed algorithm

We are now ready to formulate our **Algorithm**:

Input: A gamma-routing problem with nets $N_r = (W_r, S_r)$, $r = 1, ..., n$ satisfying conditions (i) and (ii) above.

Initialization: Relabel nets such that N_r, $r = 1, ..., u$ are the uniquely realizable ones, and fix the realization of these nets. Set $\mathcal{U} = \{u+1, ..., n\}$ and initialize the sets F_r, $r \in \mathcal{U}$ as above.

Repeat while $\mathcal{U} \neq \emptyset$

Step 1: For all $r \in \mathcal{U}$ delete all gridpoints (i,j) from F_r for which the (i,j)-realization of N_r is infeasible.

Step 2: For all $r \in \mathcal{U}$ delete all gridpoints (i,j) from F_r for which the (i,j)-realization of N_r is forbidden.

Step 3: If $F_r = \emptyset$ for some $r \in \mathcal{U}$, then STOP (NO SOLUTION), otherwise

Step 4: Choose a minimal gridpoint $(i,j) \in \bigcup_{r \in \mathcal{U}} F_r$ and let N_p, $p \in \mathcal{U}$ a net for which $(i,j) \in F_p$.

Step 5: Fix the (i,j)-realization of N_p, set $F_p = \{(i,j)\}$ and $\mathcal{U} = \mathcal{U} \setminus \{p\}$.

Stop: Output solution.

To prove the correctness of the above algorithm, first we have to verify that its steps are well defined, and it will stop either by producing a realization for all the nets, or stops with NO SOLUTION. Next, we shall prove that there are indeed no feasible solutions to the problem, when the above algorithm terminated with NO SOLUTION. Finally, we shall show that the algorithm can be implemented to run in $O(nsw)$ time.

It is quite clear that Step 3 can always be carried out in a unique way. To see that no ambiguity arise in Step 4, we need the following two lemmas.

In the first lemma below we show that in the course of the above algorithm, in Step 4, any minimal gridpoint corresponds always to a unique net as a feasible realization.

Lemma 1 *If $(i,j) \in \bigcup_{r\in\mathcal{U}} F_r$ is a minimal gridpoint, then there exists a unique index $p \in \mathcal{U}$ for which $(i,j) \in F_p$.*

Proof. Let us observe first that $(i,j) \in F_r$ implies that either $(0,j) \in W_r$ or $(i,0) \in S_r$. Thus, if there were two nets N_p and N_q such that $(i,j) \in F_p \cap F_q$, then one of them must be a West-only, and the other one must be a South-only net. Let us say, e.g. that $S_p = \emptyset$ and $W_q = \emptyset$.

Then, by the definition of an (i,j)-realization, we must have

$$j = \min W_p \text{ and } i = \min S_q.$$

Since (i,j) is minimal in $\bigcup_{r\in\mathcal{U}} F_r$, for all $(i',j') \in F_p$ we have $i' \geq i$ and $j' = j$, and thus all possible realizations in F_p of N_p would share the horizontal line segment $[(i,j),(i',j)]$ with the considered (i,j)-realization of N_q, i.e. would be forbidden and hence would have been eliminated from F_q, contradicting the fact that $(i,j) \in F_q$. This contradiction proves that no two nets can have the same minimal gridpoint in their sets of feasible realizations. $\square$

In the next lemma we show that different minimal gridpoints in Step 4 of the above algorithm correspond to non conflicting realizations, and hence fixing these realizations in any order will yield the same solution.

Lemma 2 *If $(i,j) \neq (i',j')$ are both minimal gridpoints in $\bigcup_{r\in\mathcal{U}} F_r$, and $(i,j) \in F_p$, $(i',j') \in F_q$ (then by Lemma 1 we must have $p \neq q$), then these two realizations of N_p and N_q are not in conflict.*

Proof. Without loss of generality we may assume that e.g. $i < i'$ and $j > j'$. It is easy to verify that the only possible conflict between these realizations occurs when N_q is West-only, and N_p is South-only, and either $(i',0) \in S_p$, or $(0,j) \in W_q$ (or maybe both).

Let us consider e.g. the case when $(i',0) \in S_p$. Since $(i,j) \in F_p$ is a minimal gridpoint, we have $j'' \geq j$ for all $(i,j'') \in F_p$, and hence all those realizations of N_p would share the vertical line segment $[(i',j'),(i',j'')]$, which is a non-empty segment, since $j' < j \leq j''$. Thus, the (i',j')-realization of N_q would be forbidden by the definition, contradicting the fact that $(i',j') \in F_q$. The other case, i.e. when $(0,j) \in W_q$, can be treated similarly.

Since no other case of conflict can arise, the above contradictions prove the lemma. □

The above lemmas show that in Step 4 of the **Algorithm** there is a unique net corresponding to every minimal gridpoint, and these nets can be realized in any order. To verify the correctness of the algorithm the only thing left to show is that in case of termination in Step 3, there are indeed no solutions.

Lemma 3 *If the above* **Algorithm** *stops with no solution produced, then there exists no feasible solution.*

Proof. Let us relabel the nets according to the order the **Algorithm** processed them, and assume that N_r was realized at $(i_r, j_r) \in F_r$ for $r = u+1, ..., t$ $(t < n)$, after which $F_{t+1} = \emptyset$ was found.

Let us then assume indirectly that there exists a feasible solution, and let k be the maximal index for which N_r is realized in the feasible solution by $(i_r, j_r) \in F_r$, for every $u < r < k$, and in which N_k is realized at $(i', j') \neq (i_k, j_k)$. We may assume without any loss of generality that N_k is South-only. Since (i_k, j_k) is a minimal element of F_k, according to Step 4, after the specified realizations of the first $k-1$ nets, we must have

$$i_k = i' \text{ and } j_k < j'.$$

Since $(i_k, j_k) \in F_k$ is a feasible realization of N_k at this moment, the half line (y, j_k) for $y \geq i_k$ must be unoccupied (no via hole and no horizontal segment is used) by the realizations of the first $k-1$ nets.

Let us now fix the realizations of all uniquely realizable nets and all West-only nets in the feasible solution, and let us run Gallai's algorithm for the remaining South only nets. Since there is a feasible solution, Gallai's algorithm is guaranteed to produce one, by processing the nets in left-to-right order and placing the horizontal segments of the realizations always on the first available horizontal line. Thus, by the selection of $(i_r, j_r) \in F_r$ for $r < k$, all South-only nets N_r with $r < k$ will be realized by Gallai's algorithm with exactly the same realization as in the **Algorithm**. As a next step, the horizontal segment of N_k will be placed on the horizontal line (y, j_k), $y \leq i_k$, since that line is the first available according to the observations we made above. Thus Gallai's algorithm guarantees the existence of another feasible solution, in which there is one more net, counting from the beginning, which is realized in agreement with our **Algorithm**. Repeating the above, we could arrive to a feasible solution, realizing all the nets N_r $r = 1, ..., t$ exactly the same way as the **Algorithm**, contradicting the fact that at this moment N_{t+1} cannot be realized. This contradiction shows that our assumption on the existence of a feasible solution must have been wrong, and thus proves the lemma. □

To see the complexity of the above algorithm, we have to observe first that by using appropriate data structures, steps 3–5 of the algorithm can be carried out in constant time

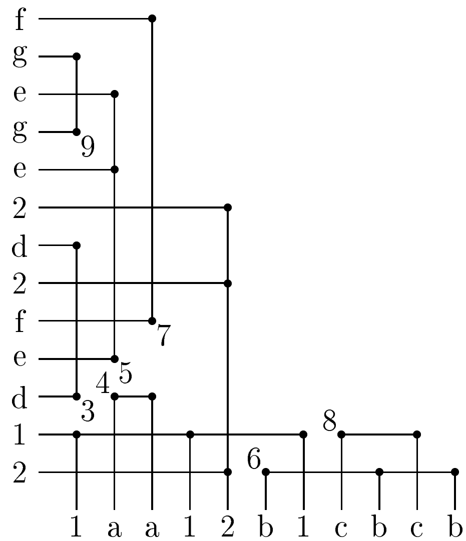

Figure 4: A solution to a simple gamma-routing problem, produced by our algorithm. The terminals of the nets are labeled by the same letter – nets 1 and 2 are uniquely realizable, nets a,b, and c are South-only, while nets d,e,f, and g are West-only. The numbers within the routing region denote the order in which the nets were realized by the algorithm.

per checked gridpoint. Since a gridpoint can correspond to at most two nets, the total time needed to perform steps 3–5 is $O(sw)$. It is also easy to see that both steps 1 and 2 can be implemented to run in $O(sw)$ time, and since there are no more than n iterations of the while-loop, the total time of all the steps of the algorithm can be bounded by $O(nsw)$.

Acknowledgement Research partially supported by the Hungarian National Foundation for Scientific Research (Grant Numbers OTKA T 17181, 17580 and 19367) and by the Hungarian Ministry of Education (Grant number FKFP 409/1997). The first author gratefully acknowledges partial support by the Office of Naval Research (grant N00014-92-J1375), by NATO (grant CRG 931531) and by NSF (grant INT 9321811). Some remarks of Éva Tardos are also gratefully acknowledged.

References

[1] GALLAI, T. Unpublished; announced in Hajnal, A. and Surányi, J.: On the Partition of Graphs into Complete Subgraphs. *Ann. Univ. Sci. Budapest Eötvös*, Vol. **1** (1958), pp. 115-123 (in German).

[2] LaPAUGH, A. S. A Polynomial Time Algorithm for Optimal Routing around a Rectangle. *Proc. 21st FOCS Symp.*, (1980), pp. 282-293.

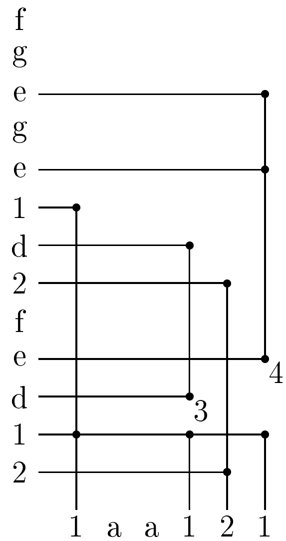

Figure 5: An unsolvable simple gamma-routing problem. After processing the uniquely realizable nets the algorithm realized the nets d and e. At this point the algorithm stops without producing any solution because F_f is empty: F_f contains gridpoints from the 5th row and the realizations $(1,5),(4,5),(5,5),(6,5)$ are infeasible and the realizations $(2,5),(3,5)$ are in conflict with the possible realizations of N_a.

[3] SZYMANSKI, T. G. Dogleg Channel Routing is NP-Complete. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. **CAD-4** (1985), No. 1, pp. 31-41.

[4] JOHNSON, D. S.: The NP-Completeness Column: an Ongoing Guide. *Journal of Algorithms*, Vol. 3, 1982, pp. 381-395.

[5] LENGAUER, Th. *Combinatorial Algorithms for Integrated Circuit Layout.* (Stuttgart, Teubner. Chichester, Wiley, 1990.)

[6] MÖHRING, R. H., WAGNER, D. and WAGNER, F. VLSI Network Design. In *Handbooks in OR & MS: Network Routing*, Vol. **8**, Elsevier, 1995, pp. 625-712.