Switchbox Routing with Movable Terminals

Jon Hamkins*
hamkins@uiuc.edu

Donna J. Brown*
djb@uiuc.edu

Coordinated Science Lab.
University of Illinois
1308 W. Main St.
Urbana, IL 61801

Abstract

We consider the Switchbox Routing Problem (SRP) in which terminals have some flexibility in placement on the border. The general problem with position constraints is NP-complete, as is the problem with only *order constraints or separation constraints. We do solve the problem for the case in which terminals are permutable within prespecified groups of adjacent vertices, called clusters. Whenever possible, our algorithm determines a terminal assignment such that the resulting SRP is solvable; the total time to assign the terminals and to construct a layout is $O(N \log N)$, where N is the number of nets. Our results extend to multiple layers and to convex grids.*

1 Introduction

In automated VLSI design, the positions of terminals on the boundary of a switchbox have traditionally been fully specified at the time the routing phase is begun (see e.g. [Fra82, MP86]). In many instances, however, there could be flexibility in the placement of the terminals. For instance, there may be logically equivalent terminals, such as the inputs of an AND gate. And for VLSI designs that contain programmable function cells such as PLAs, ROMs, and gate arrays, locations of specific terminals may be set arbitrarily along the boundary of the cell with no effect on the cell area. Current CAD tools that compute programmable cells usually allow the user to specify the desired terminal order along the boundary.

It is not surprising that allowing flexibility in the placement of terminals can lead to more efficient routing; this strategy has been studied by a number of people for the problem of minimizing channel routing area (see e.g. [WLL88, CW90, CW91, HB92a]). And [DB89] has shown that the number of vias may be reduced when terminals are allowed to move during the routing phase. In this work we are considering the problem of routing switchboxes and are concerned with the basic question of whether a switchbox problem is even routable within the given region. Unfortunately, the above techniques provide no method for converting a congested, unsolvable switchbox routing problem into a solvable one.

Certainly a switchbox too congested to wire could be expanded (perhaps doubling or even quadrupling the area) to obtain a solvable problem. This type of strategy has been used e.g. by [MP86] and [BS90]. However, such a solution would be impractical for switchboxes with fixed dimensions. As an alternative, we present an algorithm that assigns the terminals, subject to constraints, to locations along the boundary of the switchbox such that the resulting routing problem is solvable, whenever such an assignment exists. The algorithm runs in $O(N \log N)$ time, and the result also holds when the routing region is a convex grid (instead of a rectangle) or when overlap routing is allowed. We show that some more general forms of the problem are NP-complete.

2 Preliminaries

2.1 The switchbox routing problem

The well-known Switchbox Routing Problem (SRP) is the problem of constructing edge-disjoint paths between designated pairs of terminals on the boundary of a rectangle. The input to the problem consists of

*This work was supported in part by the Joint Services Electronics Program under Contract N00014-90-J-1270.

a rectangular grid and pairs of terminals along the boundary that specify nets to be electrically connected by routing within the rectangle. Wires are required to run along the vertical and horizontal edges of the grid, and no two nets' paths can share an edge. Paths are allowed to knock-knee at a vertex. See Figure 1(b).

In a SRP each net is composed of exactly two terminals, and we require the terminals to be assigned to the vertices (pin locations) along the boundary of the rectangle. We label each vertex by its (x, y) coordinates in the plane, where the lower left corner of the rectangle has coordinate (1,1). Each boundary vertex may contain 0 or 1 terminal, except that corner vertices may contain 0, 1, or 2 terminals. If a boundary vertex contains the maximum allowable number of terminals, we say that the vertex is *full*, and if it contains no terminal, we say it is *empty*. We say two nets *cross* if the line segments formed by joining each net's terminals intersect. A *cut* is a line separating the vertices of the rectangle into two sets. A *vertical cut* (*v-cut*) is a vertical line between two columns, and a *horizontal cut* (*h-cut*) is a horizontal line between two rows. We let $d(x)$ denote the *density* of cut x, the number of nets having a terminal on each side of the cut; $cap(x)$ denotes the *capacity* of cut x, the number of grid edges crossing the cut. The *free capacity* of cut x, denoted by $fcap(x)$ is given by $fcap(x) = cap(x) - d(x)$. If $fcap(x) = 0$, then x is *saturated*.

Consider the SRP in Figure 1(a). There are 7 empty boundary vertices, e.g. (1,1) and (4,6). All remaining vertices are full, except vertex (7,6) which is neither empty nor full. Net 5 crosses every net except net 4. We have indicated the v-cut between columns 5 and 6 with a dotted line. The capacity of this cut is 6 and the density is also 6, so it is a saturated cut. We have labeled the density and parity congestion (defined below) for each v-cut and each h-cut.

The *cut criterion* states that $fcap(x) \geq 0$ for every v- and h-cut x. Clearly, this is a necessary condition for solvability of the problem. It is not sufficient, however. To describe the sufficient condition, we need more terminology.

The *available degree* of a vertex is the degree of the vertex minus the number of terminals located at it. This indicates the number of wire segments that can be routed to the vertex. For example, consider the left terminal of net 5 in the SRP shown in Figure 1(a). There is one terminal located at a vertex of degree 3, so the available degree is $3-1 = 2$. Consider any v-cut x in a SRP, and suppose there are t saturated h-cuts. This implies that the part of the grid left of x may be decomposed into $t + 1$ sets of vertices, $T_1, ..., T_{t+1}$, such that for all i, $1 \leq i \leq t + 1$, no two vertices of T_i are on opposite sides of any saturated h-cut. when we consider an arbitrary h-cut in conjunction with all v-cuts. We say T_i is an *odd set* if it contains an odd number of vertices of odd available degree. A simple argument shows that the number of odd sets on each side of x is equal. We define this common number to be the *parity congestion* of x, denoted $pc(x)$.

The *revised cut criterion* states that $fcap(x) \geq pc(x)$ for every v- and h-cut x. A parity argument, first presented in [Fra82], shows that this criterion is necessary and sufficient for the solvability of a SRP.

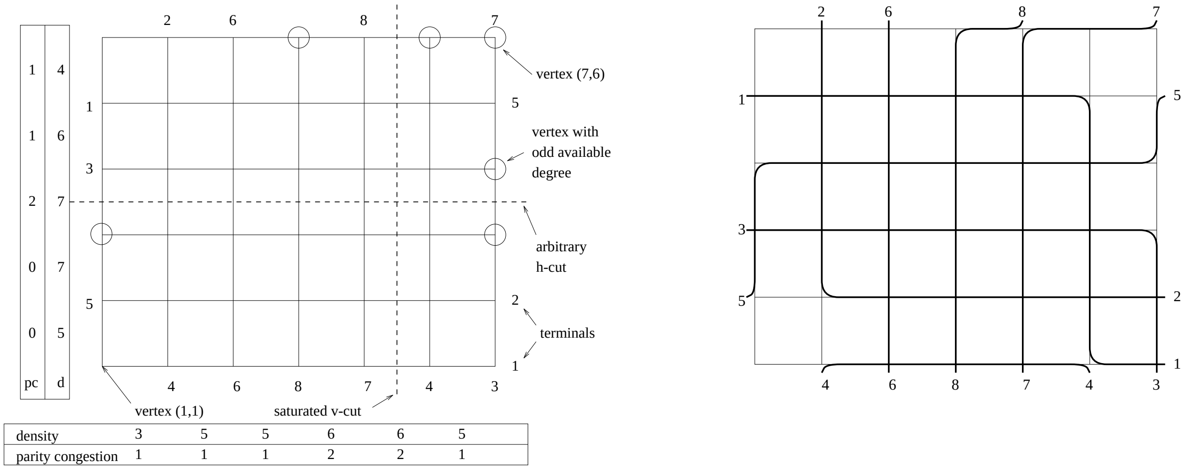
Consider our example SRP in Figure 1(a). The vertices of odd available degree have been circled. Let x be the (arbitrarily chosen) h-cut shown. A dotted line indicates the one saturated v-cut. Consider the two sets below x: one left of the v-cut, and one right of the v-cut. Each of the two sets contains 1 vertex of odd available degree, so $pc(x) = 2$. Since $fcap(x) = 7 - 7 = 0$, the revised cut criterion does not hold, and indeed the SRP is unroutable. Moving a single terminal results in the solvable SRP shown in Figure 1(b).

2.2 The terminal assignment problem

Often, a SRP does not satisfy the revised cut criterion and cannot be successfully routed. However, by taking advantage of inherent flexibilities in the placement of terminals along the boundary, the SRP might be made solvable. We now offer a description of the terminal assignment problem (TAP) for switchbox routing. As for the SRP, the input to the problem consists of a rectangular grid and N nets each with two terminals along the boundary. Position constraints are given by sets B_k of allowable terminal locations: $P = \{B_k \subseteq \{\text{boundary vertices}\} : 1 \leq k \leq 2N\}$. The kth terminal, t_k, can only be assigned to vertices in B_k.

A TAP assignment is a function f describing the assignment of terminals to vertices on each boundary of the switchbox such that the position constraints are satisfied and such that no vertex has more than the allowable number of terminals located at it, i.e. for all k, $1 \leq k \leq 2N$, we have $f(t_k) \in B_k$.

If a solution to the TAP is such that the revised cut criterion holds, we say the terminals have a *successful assignment*, indicating that the resulting SRP is solvable. Our goal is to produce a successful assignment whenever one exists.



(a) (b)

Figure 1: (a) An unroutable SRP. (b) A similar but routable SRP.

3 Solving the TAP

3.1 The general problem is NP-complete

Unfortunately, the problem of finding a successful assignment is NP-complete. In [AH87], Atallah and Hambrusch introduce a problem in which each terminal has two possible locations (i.e., $|B_k| = 2$ for all k), not necessarily adjacent, and show that the problem of minimizing density is NP-complete. Their argument was given for channel routing problems (switchboxes with no terminals on the left or right sides), but solving channel routing problems in general can be no harder than solving SRP's, since it is a special case of the SRP. Also, their work only addresses the minimization of density, not the sufficient conditions for routability, but a simple modification of their argument (for example, requiring every other vertex on the top and bottom boundary to be empty) equates density with required channel width. Therefore, the problem of finding a successful assignment of terminals on a switchbox is NP-complete.

3.2 An algorithm for a special case

Although the general problem is NP-complete, we are able to solve an important special case of the problem. Sets of adjacent border vertices called *clusters* are specified. A cluster may contain any number of vertices, but may not contain vertices on both sides of a corner vertex. Each terminal belongs to a unique cluster and may be assigned to any vertex within the cluster. That is, if two terminals t_i and t_j belong to the same cluster, then $B_i = B_j$.

We shall provide an algorithm which solves the cluster TAP and prove that the assignment is successful, whenever a successful assignment exists. Without loss of generality, we shall assume that no net N_i has both its terminals in the same cluster. If this did occur, then N_i could be trivially routed simply by placing both of N_i's terminals at adjacent vertices.

Consider a given assignment for a problem. If the assignment contains crossing nets and these nets have terminals in the same cluster, we may uncross the nets by permuting the two terminals in that cluster. It is not hard to show the following:

Lemma 1 *Uncrossing the nets does not increase the density or parity congestion of any v- or h-cut.*

We now present our algorithm, which is applied to a single cluster at a time. Due to space limitations, the presentation is informal and all proofs are omitted.

Algorithm **ASSIGN**

Step 1. Permute the terminals in the cluster such that no two nets that each have a terminal in the cluster cross.

Step 2. Slide the terminals to the counterclockwise side of the cluster, forming full vertices.

Step 3. Starting with the terminal furthest clockwise in the cluster (and then proceeding counterclockwise) slide the terminal clockwise to the vertex in the cluster which minimizes the length between it and its corresponding terminal, but never placing the terminal at a vertex that is already full.

As an example, a default assignment of terminals is shown in Figure 2(a), representing a possible "congestion-oblivious" assignment. In fact, this SRP is not solvable, for the v-cut between columns 2 and 3 has capacity 6 and density 8. However, if terminals are permutable within the indicated clusters, algorithm ASSIGN obtains the terminal assignment shown in (b), and the resulting SRP is easily routable.

Theorem 1 *ASSIGN minimizes the density of any v- or h-cut. In fact, ASSIGN produces a successful assignment for a SRP, if one exists.*

ASSIGN is an efficient algorithm, requiring only the time of a sorting algorithm, $O(N \log N)$, where N is the number of nets. Mehlhorn and Preparata [MP86] provide a $O(N \log N)$ time algorithm for constructing a layout if one exists, so we have the following:

Corollary 1 *We may determine in $O(N \log N)$ time whether a TAP has a successful assignment; if it does, then the assignment and corresponding layout take $O(N \log N)$ time to compute.*

4 More general problems

4.1 Convex grids and overlap routing

We may extend our results to regions more general than the rectangle. A grid is *convex* if any two vertices in it can be joined by a path with at most one bend (see e.g. [LS87]). We also may extend our results to new multilayer routing technologies in which k wires can be routed along the same edge of the grid. We have the following:

Theorem 2 *ASSIGN produces a successful assignment for a k-ary overlap convex grid routing problem, if one exists. A successful assignment and layout take $O(N \log N)$ time to compute.*

4.2 Order and separation constraints

As we have seen, clusters of permutable terminals model some of the flexibilities that naturally occur in VLSI design. There are other flexibilities that we have not yet mentioned. When there are several functional modules on the same side of a switchbox, the left to right order of these modules might not be fixed. For example, when two modules with the same height abut a common border of a routing region, we may interchange them. Furthermore, the separation between two adjacent modules is also variable [CW91, HB92a, She92]. These flexibilities cannot be modeled by position constraints alone. We introduce two new types of constraints.

The first type of constraint is the *order constraint*, given by $O = (\{t_1, \ldots, t_{2N}\}, \prec)$, a partially ordered set. This constraint associates a pair of terminals with the clockwise order the terminals must appear on the boundary. That is, if $t_i \prec t_j$, then a clockwise scan of the boundary starting at vertex (1,1) must encounter terminal t_i before terminal t_j. In our work, it is not necessary that the relation be defined for every pair of terminals; if it is not defined, then either order is acceptable. This constraint may model cells whose relative ordering along the boundary is fixed, but whose internal terminals may be permuted. It may also model cells which may be permuted along the boundary, but whose internal terminals have a fixed order.

The second type is the *separation constraint*, given by S, which associates with each terminal the minimum and maximum distance that a neighboring terminal may be assigned. For example, for a terminal t_i on the top boundary, if $S(t_i) = (2, 10)$, then an assignment of t_i to the vertex at column c implies that the terminal adjacent to t_i on the left must be assigned to a vertex at a column between columns $c - 10$ and $c - 2$. This constraint may model terminals which are at fixed distances from each other, such as terminals of a custom cell. It also may model the designer's desire to keep groups of wires near each other, and the need for extra space to accommodate wires of varying thicknesses.

The TAP, when subject to position and order constraints, was shown in [CW90] to be NP-complete. We improve their result by showing that the problem remains NP-complete when there are no position constraints at all; we call this the Order Constraint Problem (OCP). Finally, the problem with separation constraints is also NP-complete, even when there are no position or order constraints; we call this the Separation Constraint Problem (SCP).

Theorem 3 *OCP is NP-complete.*

Theorem 4 *SCP is NP-complete.*

5 Conclusions

We have given an algorithm that assigns the terminals to the boundary of a switchbox such that the resulting SRP is routable, whenever such an assignment is possible. The terminals have position constraints within prespecified groups of adjacent vertices. This

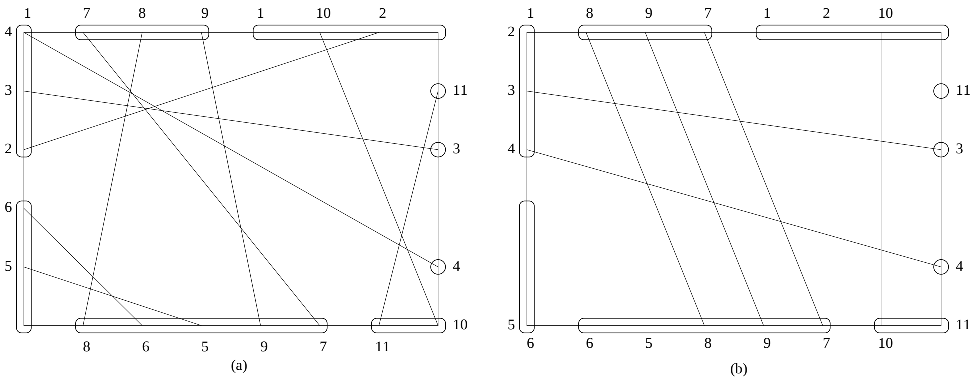


Figure 2: (a) A SRP with a default terminal assignment. Lines are drawn between terminals forming nets. (b) The problem after terminal reassignment. Note that no two nets from the same cluster cross.

is the first work we know of that relates movable terminals to the routability of a switchbox. Previous work on movable terminals has been done for channel routing, and usually involved density minimization only, whereas the problem in the switchbox requires both density and parity congestion minimization.

Position constraints more general than adjacent-vertex position constraints (i.e., clusters) result in an NP-complete problem, and we show that order constraints and separation constraints each result in NP-complete problems as well, even when there are no constraints on position. These results hold even for the channel routing problem, and even if every net has one terminal on the top boundary and the other on the bottom boundary. The algorithm and the NP-completeness results can be extended to routing in a convex grid and to overlap routing.

There are a number of open problems, including extensions to nonconvex regions and multiterminal nets.

Acknowledgement: We would like to thank Majid Sarrafzadeh for suggesting the switchbox TAP.

References

[AH87] M. J. Atallah and S. E. Hambrusch. On bipartite matching of minimum density. *J. Algorithms*, 8:480–502, 1987.

[BS90] M. Brady and M. Sarrafzadeh. Stretching a knock-knee layout for multilayer wiring. *IEEE Trans. on Computers*, 39(1):148–151, January 1990.

[CW90] Y. Cai and D. F. Wong. Optimal channel pin assignment. *IEEE Trans. Comput.*, 10(11):1413–1424, November 1991. Also appearing in *ICCAD*, pages 10-13, 1990.

[CW91] Y. Cai and D. F. Wong. Minimizing channel density by shifting blocks and terminals. In *ICCAD*, pages 524–527, 1991.

[DB89] J. S. Deogun and B. B. Bhattacharya. Via minimization in VLSI routing with movable terminals. *IEEE Trans. CAD*, 8(8):917–920, August 1989.

[Fra82] A. Frank. Disjoint paths in a rectangular grid. *Combinatorica*, 2(4):361–371, 1982.

[HB92a] J. Hamkins and D. Brown. The terminal assignment problem: Minimizing density, net span, vias, and more. Manuscript, University of Illinois, 1992.

[HB92b] J. Hamkins and D. J. Brown. Routing in a rectangle with k-ary overlap. In *Proc. Second Great Lakes Symposium on VLSI*, pages 144–151, February 1992.

[LS87] T. Lai and A. Sprague. On the routability of a convex grid. *J. Algorithms*, 8:372–384, 1987.

[MP86] K. Mehlhorn and F. P. Preparata. Routing through a rectangle. *J. ACM*, 33(1):60–85, 1986.

[She92] N. A. Sherwani. Optimal algorithms for minimizing total channel density. In *Proc. 13th Ann. Conf. on Communication, Control, and Computing*, 1992.

[WLL88] D. F. Wong, H. W. Leong, and C. L. Liu. *Simulated Annealing for VLSI Design*, chapter Permutation Channel Routing, pages 99–122. Kluwer Academic Publishers, 1988.